# The Persistence of Charm in the Relentless Decay of Beauty

Thesis by

Mark Adler

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1990

(Submitted May 24, 1990)

# Acknowledgement

I would like to first thank Mark Wise for suggesting the problem and providing helpful comment and direction during the work. I also thank Frank Porter for his assistance and for his excellent pedagogy in Ph 129 and Ph 231, Steve Frautschi for his generous supply of slack, and Donna Driscoll, just for being a nice person.

I especially thank Hughes Aircraft Company for supporting me for the duration with a Howard Hughes Doctoral Fellowship.

For support and friendship, I thank Gus Solomon (the last Freudian voice teacher in L.A.), Greg Cardell, Bill "I got a ton o' those" Wagner, and Lucy DesJardin.

# Abstract

The results of calculations of semileptonic $B_c^{\pm}$ meson exclusive decay channels using the quark potential model are presented. These results are compared with estimations made using the spectator model. The polarization of charmonia states resulting from a $b \rightarrow c$ decay are also calculated, providing a more detailed experimental check of the model.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Relentless Decays

*All beauteous things for which we live*

*By laws of space and time decay.*

*But Oh, the very reason why*

*I clasp them, is because they die.*

— William Cory 1823–1892

## 1.1 Explaining (Almost) Everything

We begin at the beginning.

In the Standard Model description of particle physics, the players are: the elementary fermions and anti-fermions; the gauge bosons which mediate the electroweak and strong forces; and the Higgs boson, which gives masses to the particles. The heuristic picture is that matter is made of the fermions, and that the fermions interact with each other, convert among themselves, and bind with each other, all through the exchange of gauge bosons.

The fermions carry integer-valued, conserved traits: charge, hypercharge, and color, that determine the respective electromagnetic, weak, and strong interactions they participate in. The corresponding gauge bosons for these interactions are the photon, the $W^{\pm}$ and $Z^0$, and the gluons. An elementary fermion can emit or absorb a gauge boson, in the process changing its momentum, and possibly its traits and character. The same process turned on its side allows the conversion of fermion–anti-fermion pairs to and from gauge bosons.

The elementary fermions are broken down into six quarks, which carry color and hence interact via the strong force, and six leptons which do not. All the fermions participate in weak interactions, and all but three (the neutrinos) carry charge and interact electromagnetically. In general, at currently accessible energies, the strong interaction will dominate if it is allowed, the electromagnetic interaction will dominate if strong is not allowed, and if neither are allowed, then and only then will weak processes be significant. In particular, decays of higher mass states into lower mass states will proceed first by strong decay channels, then by electromagnetic decay channels, and finally by weak decay channels.

The quark labels or "flavors" are up, down, strange, charm, bottom or beauty, and top or truth ($u$, $d$, $s$, $c$, $b$, and $t$), in order of increasing mass. They can also be split into two sets of three by charge, with $u$, $c$, and $t$ having charge $+2/3$, and $d$, $s$, and $b$ having charge $-1/3$. The leptons are the electron ($e$), muon ($\mu$), and tau ($\tau$), in order of increasing mass all with charge $-1$, and the chargeless and massless neutrino partners of those, the $\nu_e$, $\nu_\mu$, and $\nu_\tau$. In the Standard Model, the neutrinos can only be distinguished from each other by their weak couplings. Only the lowest mass fermions are completely stable in the model, since the weak interaction (but not the strong or electromagnetic interactions) allows flavor changing quark decays and decays of the $\mu$ and $\tau$ into electrons. This leaves only the up and down quarks, electrons, and neutrinos to make up conventional matter. It is through these relentless decays that the weak interaction component of the Standard Model can be observed.

Due to the observed and conjectured confinement of color, quarks are only seen in bound states (hadrons) of either three quarks (or three anti-quarks)—the baryons, or a quark and an anti-quark—the mesons, or combinations of these (the most prevalent being atomic nuclei). Mesons are of particular interest for experimental and theoretical investigation, since they are produced more easily than baryons, and have simpler structure and dynamics than baryons. These systems are excellent and crucial probes into the validity and limits of the Standard Model. In particular, the decays of heavy mesons explore higher energy, and hence shorter distance phenomena, and so push out the boundaries of experience and test the mettle of our theories. So far, with the notable exception of gravity, all known physical phenomena are believed to be described by, or are at least compatible with, the qualitative and quantitative predictions of the Standard Model. For various reasons, including technical and aesthetic ones, and possibly the desire of continued employment, physicists don't believe for a second that the Standard Model is the final word, and so the most urgent need is to show it to be wrong, and use the nature of the failure to point the way for new theories.

## 1.2 The Inevitable Weak Decay

Of the quarks in the model, only the first five have been seen, and the heaviest of those, beauty, has been convincingly seen only in mesons. Of the 25 possible combinations of known quarks and anti-quarks that can make up a meson, only four have not yet been seen: $b\overline{s}$, $\overline{b}s$, $b\overline{c}$, and $\overline{b}c$. There have been hints of the first two, the $\overline{B}^0_s$ and $B^0_s$ [1], but no observation of the charmed beauty mesons, the $B_c^-$ and $B_c^+$. With the promise, or perhaps hope of high-luminosity $B$-factories before the end of this decade, the production of $B_c^\pm$ mesons may become accessible. Here we are interested in predicting the dynamics of $B_c^\pm$ mesons, in particular their decays into lower mass mesons.

Once a charmed beauty meson has reached its ground state through electromagnetic interactions, the only route it has for further decay is the conversion of one of its quarks to a lower mass quark. Only the weak interaction can change flavor in this way. In particular, the emission or absorption of a $W$-boson can change the flavor between the $+2/3$ charge quarks and the $-1/3$ charge quarks via the charged weak current:

$$J^\mu = -i\frac{e}{2\sqrt{2}\sin\theta_W}\overline{u}_i\gamma^\mu(1-\gamma_5)V_{ij}d_j$$

where the index $i$ represents a mass eigenstate of one of the $+2/3$ charge quarks ($u$, $c$, or $t$) and the index $j$ represents a mass eigenstate of one of the $-1/3$ charge quarks ($d$, $s$, or $b$). $V_{ij}$ is an element of the Kobayashi-Maskawa mixing matrix, a unitary $3\times 3$ matrix that gives the relative couplings of the possible transitions between the $+2/3$ charge and $-1/3$ charge quarks:

$$V = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} .$$

The Kobayashi-Maskawa (K-M) matrix is a parameter of the Standard Model that is not predicted—it must be measured. Measuring the K-M matrix in more than one way, and for distinct applications, provides a very important check on the Standard Model. Furthermore, complex phases in the K-M matrix provide a source for CP-violation, and an accurate K-M matrix is needed to check the predictions against the observations of CP-violation.

Since the quark fields can undergo global phase changes without physical consequence, $V$ is not uniquely defined. However, given a convention for what elements of $V$ to make real, the matrix is then unique. Once this is done, $V$ is determined by four parameters and we can pick those parameters as the experimentally accessible, or at least interesting quantities: $|V_{ud}|$, $|V_{us}|$, $|V_{cb}|$, and $|V_{ub}|$. They are currently known, respectively, to accuracies of about 0.1%, 1%, 25%, and hardly at all [2]. The uncertainty of $|V_{ub}|$ is largely due to a lack of data, but the uncertainty of $|V_{cb}|$ is mostly from a lack of confidence in the theoretical predictions needed to extract $|V_{cb}|$ from the data. When more data becomes available for $|V_{ub}|$, it too will be plagued with the same theoretical uncertainties.

Aside from the intrinsic value of predicting the dynamics of a new particle, one of the reasons to calculate the decay rates of $B_c^{\pm}$ is to allow comparison with experiment to improve our estimation of the reliability of the various techniques used to arrive at those rates, and to potentially improve the models used. This in turn can tighten our constraints on the K-M matrix by improving the theoretical part of the determination at the same time that more experimental data on the matrix elements are being accumulated.

## 1.3 The Frustration of QCD

In principle, the Standard Model can predict meson decay rates and there should be no problem with theoretical uncertainties swamping experimental ones. However, we do not know how to perform such calculations. The main source of difficulty is that we don't know how to describe bound states of quarks in Quantum Chromodynamics (QCD), the description of the strong force in the Standard Model. What we do know how to do is perturbative expansions in the coupling constant, which works fine when the the coupling constant is small compared to one. However, asymptotic freedom in QCD (and other non-abelian gauge theories) [3, 4] tells us that the attractive color force that binds the quark and anti-quark in a meson is strongest when the quarks stray to the edges of their prison and weak when they are close together (the opposite of the abelian electromagnetic interaction). The binding is therefore dominated by long distance or soft gluons that have a coupling constant of order one, which is precisely the regime in which perturbative calculations are useless.

Not all is lost, however. We can still glean some things from the Standard Model before we are forced to look elsewhere. The $W$-boson emitted in the flavor changing weak decay is virtual, so it too must decay very quickly into a quark–anti-quark or lepton–anti-lepton pair. If we consider only the second case, i.e., semileptonic decays, then we can eliminate the need for dealing with bound quark states at that end of the decay. In fact, the leptonic part of the current is easily calculable within the Standard Model. In this case, the perturbative expansion (in fact, just the first term of the expansion) of the electroweak interaction is sufficient, since the $W$ can only go to a charged lepton and a neutrino, and those two can interact only weakly. Higher order weak interactions are very strongly suppressed (on the distance scales of interest) by the mass of the $W$ and $Z$.

## 1.4 Getting Halfway There

The charged weak current for the leptons is the same as for the quarks, except that the K-M matrix is replaced by an identity matrix. This is because the zero neutrino mass matrix allows rotating away all of the off diagonal elements of what would have been a leptonic K-M matrix. The $W$ propagator in the momentum range of interest is simply $-ig_{\mu\nu}/M_W^2$. This gives the transition amplitude:

$$T = i\frac{e^2}{8\sin^2\theta_W M_W^2} V_{ij}(\overline{u}_e\gamma^\mu(1-\gamma_5)v_{\nu_e})(\overline{u}_i\gamma_\mu(1-\gamma_5)d_j) \; .$$

Squaring and summing over spins, combining $e$, $\sin\theta_W$, and $M_W$ into $G_F$, the Fermi coupling constant (which is known more accurately than $\sin\theta_W$ or $M_W$

individually), gives:

$$\sum_{\text{spins}} |T|^2 = \frac{G_F^2}{2} |V_{ij}|^2 \ell^{\mu\nu} h_{\mu\nu}$$

where $\ell^{\mu\nu} = \sum_{\text{spins}} (\overline{u}_e \gamma^\mu (1 - \gamma_5) v_{\nu_e}) (\overline{v}_{\nu_e} (1 + \gamma_5) \gamma^\nu u_e)$, and $h_{\mu\nu}$ is the same for the quarks. Since we do not know the momenta of the quarks in the initial and final mesons, the form for the hadronic current, $h_{\mu\nu}$, is not useful for calculation. We can however calculate the leptonic current, $\ell^{\mu\nu}$, using the usual traceology. Ignoring the mass of the charged lepton (which restricts the results to the electron and muon in this case) gives:

$$\ell^{\mu\nu} = 8 \left( p^\mu_{\nu_e} p^\nu_e + p^\mu_e p^\nu_{\nu_e} - (p_e \cdot p_{\nu_e}) g^{\mu\nu} - i \epsilon^{\mu\nu}{}_{\alpha\beta} p^\alpha_{\nu_e} p^\beta_e \right)$$

where $p_e$ is the 4-momentum of the electron or muon, and $p_{\nu_e}$ is the 4-momentum of the neutrino. Since we don't know what $h_{\mu\nu}$ is, we will simply write down the most general Lorentz tensor that can be formed from the only available momenta on the quark end of the decay: the initial meson momentum, which we will call $p_B$, and the final meson momentum, $p_X$. However, we will cheat and make a very judicious choice of combinations of momenta to expand $h_{\mu\nu}$ in, in anticipation of the final answer where some of the combinations will drop out completely. In particular, we will define $a = p_B + p_X$ and $b = p_B - p_X$. We then write the hadronic current as:

$$h_{\mu\nu} = -\alpha g_{\mu\nu} + \beta_{++} a_\mu a_\nu + \beta_{+-} a_\mu b_\nu + \beta_{-+} b_\mu a_\nu + \beta_{--} b_\mu b_\nu + i \gamma \epsilon_{\mu\nu\rho\sigma} a^\rho b^\sigma$$

where $\alpha$, $\gamma$, and the $\beta$'s are functions of the available Lorentz scalars. We will also define $c = p_e$, and then since $p_B = p_X + p_e + p_{\nu_e}$, we can replace $p_{\nu_e}$ by $(b - c)$, giving for the leptonic tensor:

$$\ell^{\mu\nu} = 8 \left( b^\mu c^\nu + c^\mu b^\nu - 2 c^\mu c^\nu - (b \cdot c) g^{\mu\nu} - i \epsilon^{\mu\nu}{}_{\alpha\beta} b^\alpha c^\beta \right) .$$

Contracting the leptonic and hadronic tensors gives:

$$\begin{aligned}\ell^{\mu\nu}h_{\mu\nu} &= 8\Big(2\alpha(b\cdot c)+\beta_{++}(2(a\cdot b)(a\cdot c)-2(a\cdot c)^2-a^2(b\cdot c))+ \\ &\quad \gamma\left(2b^2(a\cdot c)-2(a\cdot b)(b\cdot c)\right)\Big)\ .\end{aligned}$$

Only one of the four $\beta$'s survived the contraction. The other three $\beta$ terms are all proportional to $c^2$, the invariant mass of the charged lepton (an $e$ or $\mu$) which is being treated as massless here. If the $\tau$ were to be included, whose mass is of the same order as the charm quark, those terms would have to be kept.

Counting kinematic variables, a decay of one particle into three particles gives us four 4-momenta or 16 numbers to start with. We subtract four for the invariant masses of the particles and another four for energy-momentum conservation, leaving eight. Picking the initial meson rest frame with three boosts leaves five. The initial meson is spinless, so we can subtract two to put the final meson on, say, the $z$-axis, and then one more rotation about the $z$-axis to put the electron (or muon) and neutrino in the $xz$ plane, leaving two kinematic variables to define the allowed phase space of the decay. We pick two dimensionless Lorentz scalars as the kinematic variables: $x=(p_e\cdot p_B)/m_B^2$ and $y=(p_B-p_X)^2/m_B^2$.

In the $B$ rest frame, $x=E_e/m_B$ and $y=1+m_X^2/m_B^2-2E_X/m_B$. It is clear in this frame that relative momenta of the quarks in the $B$ and $X$ mesons depend on $y$, but not on $x$. Given a fixed $y$, $x$ simply determines the angle of the charged lepton to the $X$-meson. Therefore the functions $\alpha$, $\beta_{++}$, and $\gamma$ that determine the hadronic current can depend only on $y$. Defining

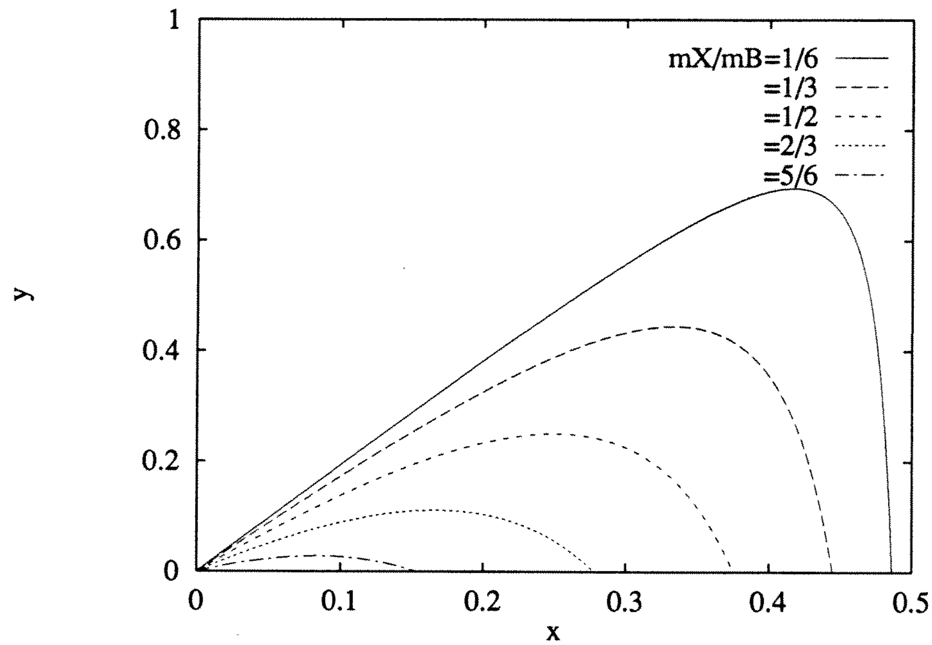



Figure 1.1: Phase space in terms of $x = E_e/m_B$ and $y = (p_B - p_X)^2/m_B^2$ for representative values of $m_X/m_B$.

the mass ratio $z = m_X^2/m_B^2$, the phase space in $x$ and $y$ is bounded in $x$ by $x > 0$ and $x \leq (1-z)/2$ and then for fixed $x$, $y$ is bounded by $y \geq 0$ and $y \leq 2x(1 - z/(1-2x))$. The maximum value for $y$ occurs at $x = (1-\sqrt{z})/2$ and gives the limit $y \leq (1-\sqrt{z})^2$. At maximum $y$, $E_X$ is at its minimum, $E_X = m_X$, so the $X$-meson is at rest in $B$ rest frame and both $E_e$ and $E_{\nu_e}$ are $(m_B - m_X)/2$, evenly splitting the available kinetic energy and going in opposite directions. Figure 1.1 shows the phase space for representative values of $m_X/m_B$. The strong suppression for large $m_X$ can be clearly seen in the relative areas of the total phase space.

We then express all of the possible inner products of $a$, $b$, and $c$ in terms of $x$, $y$, and $z$:

$$\begin{aligned}
a \cdot b &= m_B^2(1-z) \\
a \cdot c &= m_B^2(2x - \frac{1}{2}y) \\
b \cdot c &= m_B^2(\frac{1}{2}y) \\
a^2 &= m_B^2(2+2z-y) \\
b^2 &= m_B^2 y \\
c^2 &= 0\,.
\end{aligned}$$

Replacing the inner products in the contraction of the leptonic and hadronic tensors, including $|V_{ij}|^2 G_F^2/2$ for the sum of the transition amplitudes squared, using Fermi's Golden Rule, and integrating over the delta functions, we arrive at last at the formula for the differential decay rate:

$$\begin{aligned}
\frac{d^2\Gamma}{dx\,dy} &= \frac{G_F^2 m_B^5}{32\pi^3}|V_{ij}|^2\Bigg(\frac{1}{m_B^2}\alpha y + 2\beta_{++}\left(2x - y - 2xz + 2xy - 4x^2\right) - \\
&\quad \gamma y\Big(1 - z - 4x + y\Big)\Bigg)
\end{aligned}$$

where $\alpha$, $\beta_{++}$, and $\gamma$ are as yet undetermined functions of $y$, and $V_{ij}$ is the appropriate K-M matrix element for the flavor change of the initial to final meson.

Now all that remains is to determine the functions $\alpha(y)$, $\beta_{++}(y)$, and $\gamma(y)$, trying to incorporate into the weak current the momenta of the initial and final quarks from the structure of the initial and final mesons.

# Chapter 2

# Concocted Quarks

*"Then we have to look at experiment and find out what it is. This is* $+2.79$*. This is* $-1.93$ *... Hey, hey, hey, hey, hey ...*
*Those goddamn quarks are in there! I don't know what the hell was holding Gell-Mann up. This is easy. And so he kept talking about it 'behaves as if' it's made outta quarks. You're damn right it does! Cause it's made outta quarks."*
— R. P. Feynman just after deriving the magnetic moments of the proton and the neutron on the blackboard as $+3$ and $-2$ from a quark model with quarks $1/3$ the mass of the proton.

## 2.1 Encumbered Quarks

Attempts at using perturbative methods to determine the hadronic current appear to be doomed to failure, especially for exclusive decay rates to low-lying states, precisely what is needed to extract the K-M matrix elements from experiment [5]. Until significant (and I'd venture to say, unexpected) advances in non-perturbative calculations in the Standard Model are made, we must look elsewhere for some means of describing bound states of quarks, and in particular, mesons. To get a handle on the structure of mesons, we look to the surprising achievements of constituent quark models. A constituent quark has all of the characteristics of the corresponding Standard Model "current" quark, except for its mass. The constituent quark mass is roughly 300 to 500 MeV more than the corresponding current quark mass (the mass generated by the Higgs coupling) so as to account for the additional momentum carried by the incalculable soft gluons. These masses are then variable parameters in phenomenological models that generally have features inspired by both QCD and observation.

A simple example of the use of constituent quark masses is to choose the $u$ and $d$ masses as one-third of the nucleon mass, and then use these constituent quarks, with the charges and spins of the respective current quarks, to compute the magnetic moments of the proton ($uud$) and neutron ($udd$). We need to know the spin wave functions to get the magnetic moments, and it is here that QCD comes into play. The requirement of an anti-symmetric color wave function determines the spin and flavor wave functions, and gives magnetic moments of $+3\ e/2m_p$ for the proton and $-2\ e/2m_p$ for the neutron, to be compared with the experimental values of $+2.79$ and $-1.91$. If we had used the current quark masses (five to seven MeV), we would have been off by about two orders of magnitude. Lowering the constituent quark mass for the $u$ and $d$ slightly brings the prediction into closer agreement, and then those masses can be used to compute the magnetic moments of the $\Lambda$, $\Sigma^+$, $\Sigma^-$, and $\Xi^0$ in surprising agreement with experiment, at least for such a simple model [6].

Similar successes are achieved in using a phenomenological spin-spin interaction to get the mass splittings in the baryon octet and decuplet [7]. More impressive conquests, however, are seen in the bound heavy quark systems, $c\bar{c}$ and $b\bar{b}$: the quarkonia, so named because of the analogy to positronium, the quasi-bound system of an electron and positron. About the time of the discovery of the $J/\psi$ (the lowest spin 1 $c\bar{c}$, or charmonium state), Applequist and Politzer suggested a spectrum for $c\bar{c}$ states using the color Coulomb force of short-distance QCD in non-relativistic bound states [8]. In fact, the spectrum of charmonium corresponds quite well to the structure of positronium, save that the splittings in charmonium are much larger relative to the mass of the states, as is predicted from the larger coupling constant.

## 2.2 The Irresistible Attraction of Quarks

The simple attractive Coulombic potential from the one-gluon diagram in QCD, $V(r) = -4\alpha_S/3r$ is not adequate for a non-relativistic treatment of quarkonia, since it ignores confinement and leads to unrealistically large meson radii. The simplest modification is to add a term to the potential that increases with distance, and so would dominate for large distances, but goes to zero at short distances to allow the asymptotically free behavior of the $\alpha_S(r)/r$ term to dominate. Any term like $r^n$, for $n$ positive would do, such as a linear potential, $V \sim r$, or an harmonic oscillator potential, $V \sim r^2$. Predicting

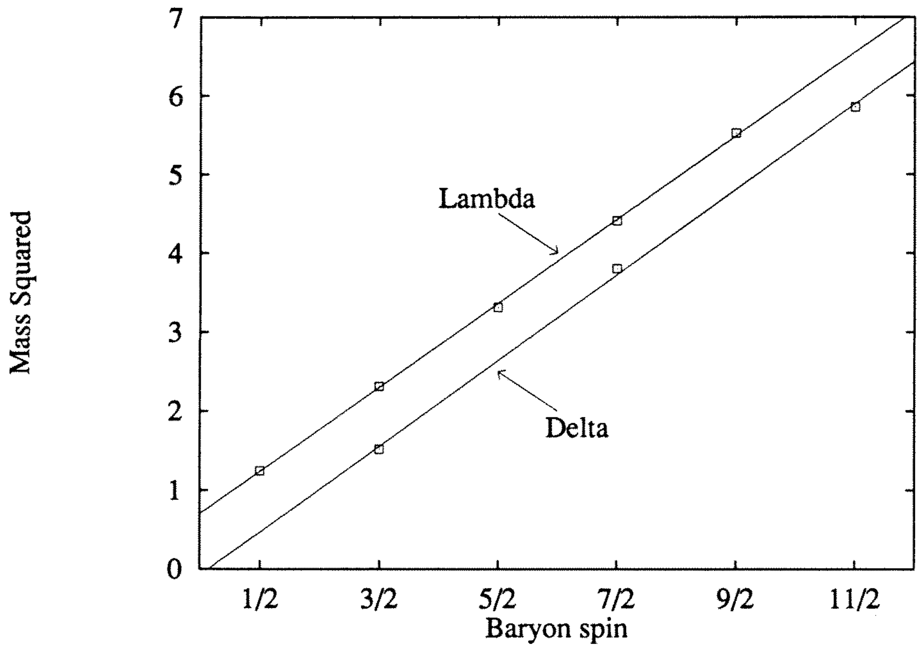



Figure 2.1: Chew-Frautschi Plot for $\Lambda$ and $\Delta$ Baryons. The $y$ axis is in units of $GeV^2$.

quarkonia levels does not distinguish one from the other, since the two potentials don't differ much over the distances the quarks are allowed to stray from each other.

There is, however, motivation from a combination of QCD and a striking observation for selecting the linear form for a confining potential [9]. The observation is that there is a linear dependence of the mass squared of baryon and meson states on the spins of those states. In Figure 2.1, the mass squared of several resonances of the $\Delta$ (isospin 3/2) and $\Lambda$ ($uds$) baryons is shown against the spins of those resonances. The slope of the lines is about 1.07 $GeV^2$.

Since gluons themselves carry color, they strongly interact with each other (attractively it turns out) and one arrives at a picture of the color lines of force being pulled into a tube or string of gluon flux connecting the two quarks. If we consider two massless quarks at the ends of such a glue tube, and let them rotate at the speed of light, then we can calculate, relativistically, the dependence of the angular momentum on the energy and get for a potential of the form $V = br^n$ (where we view $V$ as the density of the string as a function of the distance from its center), the relation $J \sim E^{(1+1/n)}$. For the observed behavior in the $\Delta$ and $\Lambda$ baryons of $J \sim E^2$, we are led to the potential $V = br$, and can even calculate $b$ to be the slope over $2\pi$. For the observed slope of 1.07 GeV$^2$, we get $b = 0.17$ GeV$^2$.

Quite similar values are arrived at by adjusting $b$ in quarkonia models with a linear confining potential to fit the observed spectra. For example $b = 0.18$ GeV$^2$ [10], and $b = 0.20$ GeV$^2$ [11]. The value of $b$ has an interesting macroscopic interpretation: the consequent force law $F = b$ says that no matter how far apart two quarks are, they are drawn together with a force of 0.17 GeV$^2$, or in more conventional units, 16 tons of force. Of course, this statement is a bit misleading, since the potential has only been shown to be faithful to reality in systems where the quarks cannot separate by more than one or two fermis ($10^{-15}$ m). Furthermore, when quarks are separated by much more than that, the color flux tube becomes massive enough to produce a quark and an anti-quark in the middle and thereby break into two flux tubes freely moving away from each other. Nevertheless, the image of a 16 ton weight helps one appreciate just how confined quarks are.

To describe fully the detailed structure of quarkonium spectra, terms must be added to the potential to describe spin-spin and spin-orbit couplings, and in some cases, a term for quark–anti-quark annihilation. The use of relativistic kinematics and expansion in powers of the momentum over the mass [12] help make up for the non-relativistic nature of the potential. When all of this is done with great care and attention to detail [10], such a model describes not only the spectra of the $c\bar{c}$ and $b\bar{b}$ systems, but for *all* mesons from the pion on up! The same set of parameters applied to the much richer spectra of baryons also gives very good agreement with observation [13]. It is these and continued herculean efforts by Isgur and associates that the calculations in this work are based on.

## 2.3 Spectator Quarks

The simplest and most direct approach to computing semileptonic decay rates of heavy mesons is to simply ignore the other "spectator" quark in the meson and treat the decaying quark as free. This free quark decay rate is calculable in the Standard Model, and gives a good first estimate for inclusive rates. However, it ignores mesons, and so ignores exclusive channels to definite mass final states. This is a serious deficiency for the lepton endpoint spectrum, since this range is dominated by the lowest mass final mesons, making the spectrum in that range strongly dependent on sharp kinematic limits and phase space restrictions that the free quark decay model is lacking. The free quark decay model is also very strongly dependent on the mass of the decaying quark, because it enters to the fifth power in the decay rate. Since both current and constituent quark masses have large uncertainties, and since either can be argued as appropriate for the free quark decay model, the possible decay rates can have quite a range and the free quark decay model loses much of its predictive capability.

Many patches are possible to improve the free quark decay model for the endpoint spectrum. Altarelli et al. [14] give the spectator quark definite mass and treat the decaying quark as a virtual particle whose invariant mass is determined by a gaussian momentum distribution for the spectator quark and a mass of the initial meson. This helps get some, but not all of the kinematics correct, since the small set of possible final meson masses is not considered. What it does take into account is the phase space available by setting the mass of the spectator quark appropriately. Their approach also leads to a divergence from QCD corrections right at the endpoint that must then be regularized.

Körner and Schuler match the free quark decay behavior at $y = 0$ to set the form factors [15], similar to the $\alpha$, $\beta_{++}$, and $\gamma$ functions defined earlier. Then the $y$ dependences of the form factors are poles in $y$, whose power is set, based on QCD counting rules, and whose location is a parameter of the model. This method can handle the kinematics, but is still does not take into account the structures of the initial and final mesons, except for a global "overlap mismatch factor" of 0.7, taken from the work of Wirbel, Stech, and Bauer [16].

Wirbel, Stech, and Bauer [16] also use pole-dominated form factors, where the pole is at the mass of a meson made of the initial and final quark in the free decay (with one of them charge conjugated, of course). Here, the behavior at $y = 0$ is set by using relativistic harmonic oscillator wave functions for the mesons, where the average transverse quark momentum, $\omega$, is a parameter of the model. They cannot fix this parameter through other means, so they present results for several different values of $\omega$. Of those mentioned so far, this one seems the most satisfying, since it at least tries to take into account the meson structure. However, the meson wave function is simply an ansatz with a free parameter, there is no motivation for the meson structure from QCD, and the wave functions are not checked against other observations.

## 2.4 A Model with Relative Potential

Isgur, Scora, Grinstein, and Wise [17] use non-relativistic harmonic oscillator wave functions that are variational solutions of the Schrödinger equation with the Coulomb plus linear quark-quark potential introduced earlier. These wave functions are the ones that were used to extend the success of quarkonia and go on to describe all the meson spectra from the $\pi$ to the $\Upsilon$. In that case, a more complicated potential was used, but here the potential is only needed to determine the meson radius, so terms beyond the Coulomb term and confining linear term do not affect the results.

Once the radii are determined, the final meson momentum wave function is boosted, and that is used with the initial meson momentum wave function directly to compute the hadronic weak current between the initial and final mesons with the dependence of the current on the boost arising naturally. This approach avoids the ad hoc pole dominance ansatz for the form factor $y$

dependence. In addition, the QCD motivated potential is used to set the meson radii instead of it being a model parameter, and the meson wave functions have already shown their effectiveness in determining meson spectra. This model is the clear choice, and is the one used here to calculate the exclusive semileptonic decays of charmed beauty mesons[1].

Du and Wang [18] chose to use the methods of Wirbel et al. to compute semileptonic decay rates of $B_c^{\pm}$. Their results will be compared with the results obtained in this work in the next chapter.

[1]It should not be overlooked, however, that one of the authors of [17] is Your Humble Servant's advisor.

# Chapter 3

# The Persistence of Charm

*"Graduate students shouldn't be jumping off mountains. Graduate students should eat, sleep, and do Physics."*
— Mark B. Wise, admonishing a wayward graduate student, and perhaps only reluctantly including eating and sleeping as valid activities.

## 3.1 The Race

Here we calculate the expected semileptonic decay rates into exclusive channels of $B_c^{\pm}$ using the quark potential model. This model should work very well for $B_c^{\pm}$, since a meson having only heavy quark flavor numbers more closely approximates a non-relativistic system of constituent quarks. The detailed predictions can then be checked against experiment to validate and possibly adjust the model.

This system provides a very good check on the interpretation of the decay of a meson as the decay of one of the constituent quarks. The reason is that for $B_c^{\pm}$, the decay of the $b$-quark is easily distinguished from the decay of the $c$-quark, and the rates of the two decays are of the same order of magnitude. In particular, considering the dominant decays, the $b \rightarrow c$ decay is suppressed by the Kobayashi-Maskawa matrix element $|V_{cb}|$ by about the same amount the $c \rightarrow s$ decay is suppressed by phase space. For example, $|V_{cb}|^2/|V_{cs}|^2$ is about $2 \times 10^{-3}$, and $m_c^5/m_b^5$ is about $6 \times 10^{-3}$, using typical values for these parameters. A more complete free quark decay calculation also gives a $b \rightarrow c$ to $c \rightarrow s$ ratio of 1/3, using the same constituent quark masses and Kobayashi-Maskawa matrix elements used in this calculation, which are listed below.

As will be seen, the treatment of the meson as a whole in the quark potential model also predicts the $b \rightarrow c$ and $c \rightarrow s$ rates to be the same order of magnitude, but unlike the free quark decay model, it predicts the $b \rightarrow c$ rate to be the larger of the two. It is clear who wins the race, since a convergent calculation would give $b \rightarrow c$ decays an even bigger lead.

## 3.2 The Calculation

This calculation is an extension of a previous calculation of semileptonic $B$ and $D$ decays in the quark potential model [17]. The predictions here are for $B_c^+ \rightarrow X^0 e^+ \nu_e$, where $X^0$ is a meson with quark composition $c\bar{c}$, $c\bar{u}$, $\bar{b}s$, or $\bar{b}d$. The decay rates of $B_c^-$ are equivalent (with the appropriate transformation of the final state) up to the accuracy of the parameters used in the calculation. The results also apply to $B_c^+ \rightarrow X^0 \mu^+ \nu_\mu$, since the muon mass can be neglected.

Table 3.1: Meson Masses. The states are labeled by $n^{2S+1}L_J$, where $n$ counts from 1 for states of a given $L$. All masses are in GeV/$c^2$.

| | $1^1S_0$ | $1^3S_1$ | $1^3P_0$ | $1^1P_1$ | $1^3P_1$ | $1^3P_2$ | $2^1S_0$ | $2^3S_1$ |
|---|---|---|---|---|---|---|---|---|
| $c\bar{u}$ | 1.86 | 2.01 | 2.40 | 2.44 | 2.49 | 2.50 | 2.58 | 2.64 |
| $c\bar{c}$ | 2.98 | 3.10 | 3.42 | 3.51 | 3.51 | 3.56 | 3.59 | 3.69 |
| $\bar{b}d$ | 5.28 | 5.33 | 5.75 | 5.80 | 5.80 | 5.80 | 5.90 | 5.93 |
| $\bar{b}s$ | 5.39 | 5.45 | 5.83 | 5.88 | 5.88 | 5.88 | 5.98 | 6.01 |
| $\bar{b}c$ | 6.27 | | | | | | | |

As in the previous calculation, only the rates to the 1S, 1P, and 2S states of the possible $X^0$ mesons are calculated. This proves sufficient for providing a reliable total rate for the $c$-quark decays, but not for the $b$-quark decays. However, to detect such decays, cuts on the electron energy would select out the low lying final states of $b$-quark decays. The calculations here cover such a range for the $b$-quark decays.

The masses used for the charmonium final states of the possible $X^0$'s (except for the $1^1P_1$ state), and the 1S $c\bar{u}$ ($D^0$ and $D^{*0}$) and $\bar{b}d$ ($B^0$ and $B^{*0}$) states are from experiment [2]. The remaining masses are from previous calculations using the quark potential model [10].

The constituent quark masses used for the mock-meson states were 0.33 GeV for the $u$ and $d$, 0.55 GeV for the $s$, 1.82 GeV for the $c$, and 5.12 GeV for the $b$. The quark-quark potential used to determine the $\beta$'s is the Coulomb plus linear potential

$$V(r) = -\frac{4\alpha_S}{3r} + br$$

where $\alpha_S = 0.5$ for $\bar{b}s$, $\alpha_s = 0.4$ for $c\bar{c}$ and $\bar{b}c$, and $b = 0.18$ GeV$^2$. The harmonic oscillator wave function $\beta_S$ and $\beta_P$ values for the $B_c^+$ and $X^0$'s were chosen to minimize the ground state energies in those sectors. The $\beta$'s then determine the meson wave functions used in the remainder of the calculation. The energy of the state is given by:

$$E = \int_0^\infty dr \left( -\frac{1}{2\mu} R^* \frac{d}{dr} \left( r^2 \frac{dR}{dr} \right) + R^* R \left( \frac{\ell(\ell+1)}{2\mu} + r^2 V(r) \right) \right)$$

where $R$ is the (properly normalized) radial part of the wave function, $\ell$ is the orbital angular momentum of the angular part of the wave function, and $\mu = 1/(1/m_p + 1/m_q)$ is the reduced mass of the system of the quarks $p$ and $q$. For the 1S states, $R \sim \beta^{3/2} e^{-\beta^2 r^2/2}$. Using the Coulomb plus linear potential we get that $dE/d\beta$ is proportional to:

$$\beta^3 - \frac{16\alpha_S \mu}{9\sqrt{\pi}} \beta^2 - \frac{4b\mu}{3\sqrt{\pi}}$$

The real root of this cubic in $\beta$ gives the minimum energy ground state, where the real root for a cubic of this form is most simply expressed as the root of $x^3 - 3ax^2 - 2b = 0$, which is $x = a + c + a^2/c$ where $c = (a^3 + b + (2a^3 b + b^2)^{1/2})^{1/3}$. Similarly for the 1P ground state, we have $R \sim \beta^{5/2} r e^{-\beta^2 r^2/2}$, which gives $dE/d\beta$ proportional to:

$$\beta^3 - \frac{32\alpha_S \mu}{45\sqrt{\pi}} \beta^2 - \frac{16b\mu}{15\sqrt{\pi}}$$

The resulting $\beta_S$ and $\beta_P$ values are shown in Table 3.2.

Since 1S and 2S states are considered here, one might expect that mixing those states would give a better fit to the potential. However, for a potential of this form and harmonic oscillator wave functions, there is exactly zero mixing of the 1S and 2S states in the ground state. (If 3S states were considered, then

Table 3.2: Harmonic Oscillator $\beta$ Values in GeV

| Parameter | $c\bar{u}$ | $c\bar{c}$ | $\bar{b}d$ | $\bar{b}s$ | $\bar{b}c$ |
|---|---|---|---|---|---|
| $\alpha_S$ | 0.5 | 0.4 | 0.5 | 0.5 | 0.4 |
| $\beta_S$ | 0.39 | 0.65 | 0.41 | 0.51 | 0.81 |
| $\beta_P$ | 0.34 | 0.52 | 0.35 | 0.41 | |

there would be mixing of the 1S and 3S states in the ground state.) The zero mixing can be seen by looking at the Hamiltonian matrix for the 1S and 2S states:

$$\begin{pmatrix} \frac{9\sqrt{\pi}\beta^3-32\alpha_S\beta^2\mu+24b\mu}{12\sqrt{\pi}\beta\mu} & \frac{9\sqrt{\pi}\beta^3-16\alpha_S\beta^2\mu-12b\mu}{6\sqrt{6\pi}\beta\mu} \\ \frac{9\sqrt{\pi}\beta^3-16\alpha_S\beta^2\mu-12b\mu}{6\sqrt{6\pi}\beta\mu} & \frac{63\sqrt{\pi}\beta^3-80\alpha_S\beta^2\mu+108b\mu}{36\sqrt{\pi}\beta\mu} \end{pmatrix}$$

The upper right element, $\langle 1\mathrm{S}|H|2\mathrm{S}\rangle$, is proportional to the derivative with respect to $\beta$ of the upper left element, $\langle 1\mathrm{S}|H|1\mathrm{S}\rangle$. This means that when the ground state energy is minimized by varying $\beta$, the matrix is diagonalized at the same time. A more careful analysis where the matrix is diagonalized first and then the ground state energy minimized gives the same result.

The relevant absolute values of the Kobayashi-Maskawa matrix elements used are $|V_{cb}| = 0.0480$, $|V_{ub}| = 0.0053$, $|V_{cs}| = 0.9744$, and $|V_{cd}| = 0.2197$. These values are only used to compare rates for the different channels. As mentioned before, the actual values for $|V_{cb}|$ and $|V_{ub}|$ are not well known.

## 3.3 The Results

The results of the calculation are shown in Table 3.3 and in Fig. 3.1, Fig. 3.2, Fig. 3.3, and Fig. 3.4.

Table 3.3: Rates for the exclusive decay channels in units of $10^{10}\mathrm{s}^{-1}|V_{qp}|^2$ where $V_{qp}$ is the appropriate Kobayashi-Maskawa matrix element. The last row includes the approximate K-M matrix elements and is in absolute units of $10^{10}\mathrm{s}^{-1}$.

| State | $c\bar{c}$ | $c\bar{u}$ | $\bar{b}s$ | $\bar{b}d$ |
|---|---|---|---|---|
| $1^1S_0$ | 750. | 234. | 2.65 | 3.04 |
| $1^3S_1$ | 2740. | 929. | 6.58 | 7.77 |
| $1^3P_0$ | 94.3 | 48.9 | 0.0137 | 0.0251 |
| $1^1P_1$ | 317. | 574. | 0.0553 | 0.172 |
| $1^3P_1$ | 108. | 554. | 0.0246 | 0.131 |
| $1^3P_2$ | 181. | 96.2 | 0.00191 | 0.00392 |
| $2^1S_0$ | 170. | 794. | 0.0144 | 0.0680 |
| $2^3S_1$ | 230. | 807. | 0.00850 | 0.0401 |
| Total rate | 4590. | 4040. | 9.35 | 11.3 |
| Absolute rate | 10.6 | 0.1134 | 8.88 | 0.543 |

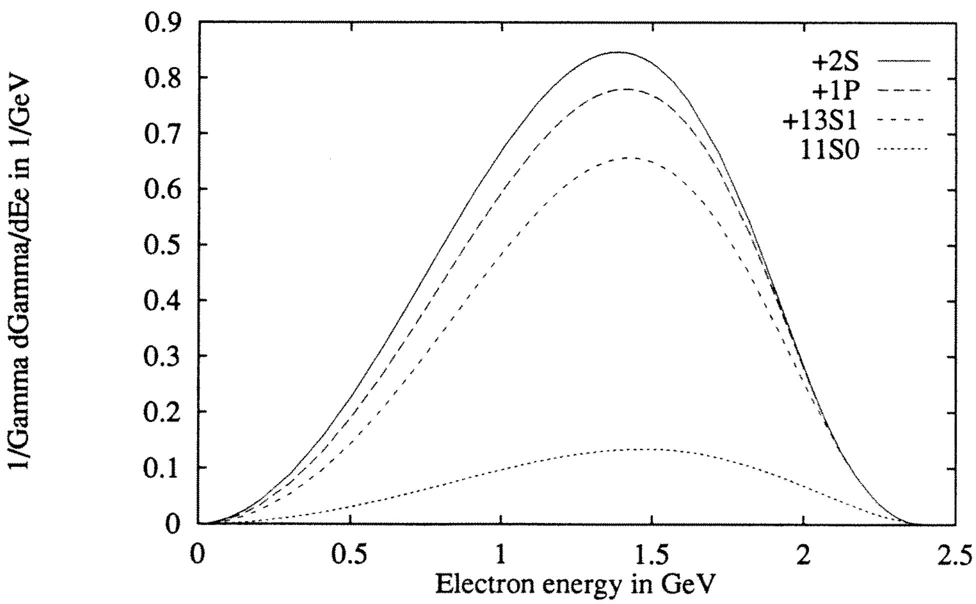



Figure 3.1: Decays of $B_c^+$ to $\eta_c$, $\psi$, and $\chi_c$ states ($\bar{b} \to \bar{c}$).

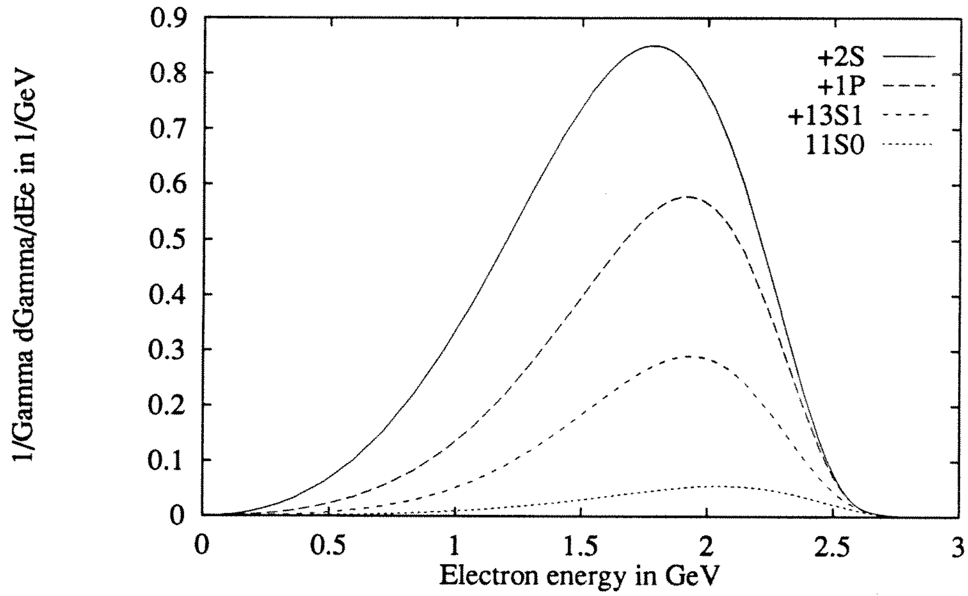



Figure 3.2: Decays of $B_c^+$ to $D^0$ states ($\bar{b} \to \bar{u}$).

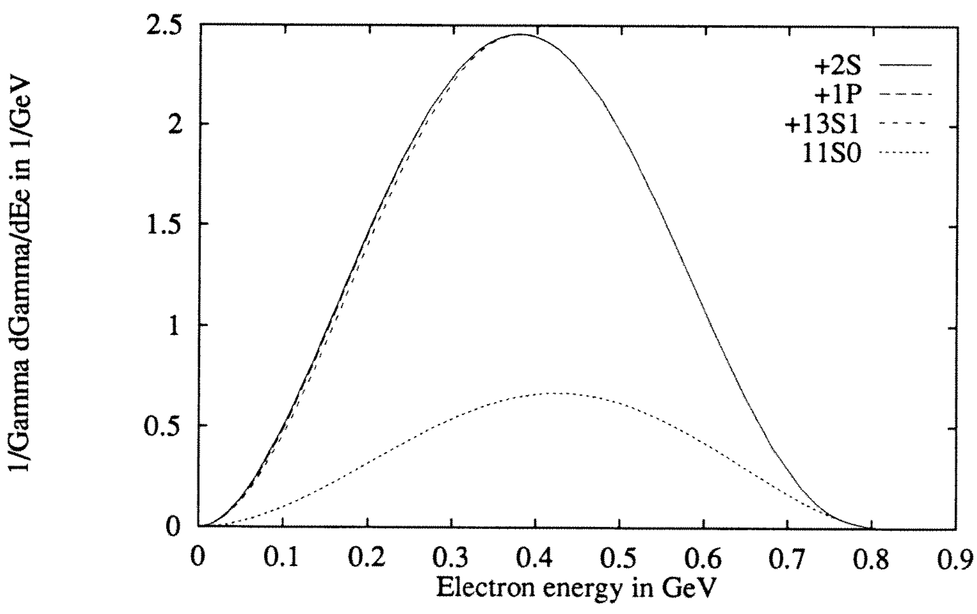



Figure 3.3: Decays of $B_c^+$ to $B_s^0$ states ($c \to s$).

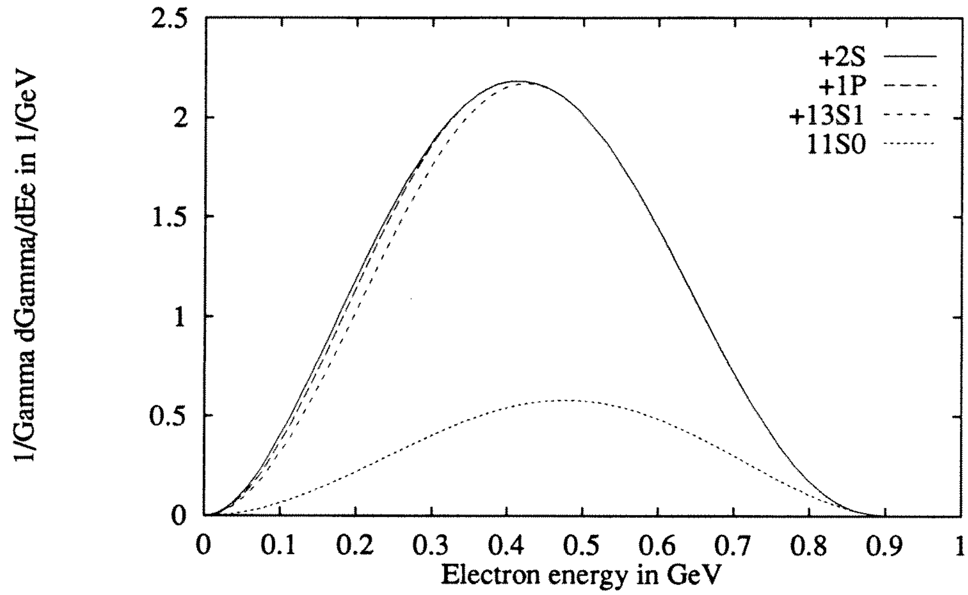



Figure 3.4: Decays of $B_c^+$ to $B^0$ states ($c \to d$).

It is clear from both the table and the figures that the $c$-quark decays converge rapidly, but that the $b$-quark decays do not. To obtain a total rate for the $b$-quark decays would require a further extension of the calculation to include many more excitation states of the charmonia and $D$ mesons. This would also add more uncertainty to the calculation, since there is little experimental data with which to check the quark potential model in this regime. However, if experimental cuts are made on the electron or muon energy, then the higher excitation states can be kinematically excluded, allowing a direct comparison with these calculations, without any extension beyond the 2S states.

It is clear that the $b$-quark decay rate and the $c$-quark decay rate are of the same order of magnitude, and in fact are nearly equal, up to the included final states. This calculation disagrees with the simpler free quark decay model

Table 3.4: Decay rates as a function of the $B_c^{\pm}$ mass. The decay rates are in units of $10^{10}s^{-1}$.

| $B_c^{\pm}$ mass | $c\bar{c}$ | $c\bar{u}$ | $\bar{b}s$ | $\bar{b}d$ |
|---|---|---|---|---|
| 6.12 GeV/$c^2$ | 9.01 | 0.107 | 3.78 | 0.266 |
| 6.15 GeV/$c^2$ | 9.32 | 0.109 | 4.56 | 0.310 |
| 6.18 GeV/$c^2$ | 9.62 | 0.110 | 5.45 | 0.360 |
| 6.21 GeV/$c^2$ | 9.94 | 0.111 | 6.46 | 0.415 |
| 6.24 GeV/$c^2$ | 10.26 | 0.112 | 7.60 | 0.476 |
| 6.27 GeV/$c^2$ | 10.58 | 0.113 | 8.88 | 0.543 |
| 6.30 GeV/$c^2$ | 10.91 | 0.114 | 10.31 | 0.617 |
| 6.33 GeV/$c^2$ | 11.25 | 0.116 | 11.90 | 0.699 |
| 6.36 GeV/$c^2$ | 11.59 | 0.117 | 13.66 | 0.787 |
| 6.39 GeV/$c^2$ | 11.94 | 0.118 | 15.59 | 0.884 |
| 6.42 GeV/$c^2$ | 12.30 | 0.119 | 17.72 | 0.990 |

which predicts the total $c$-quark decay rate to be roughly three times the total $b$-quark decay rate. Continuing the $b$-quark rate calculation to convergence would only increase the disparity. This is therefore a very distinctive prediction of the quark potential model. The relative $b$ and $c$-quark decay rates are, of course, very sensitive to $|V_{cb}|$.

Besides the K-M matrix elements, these calculations are quite sensitive to the $B_c^{\pm}$ mass, which is not known. Presumably, the mass will be known before any inclusive or exclusive decay rates are measured. In anticipation of this, the calculation was repeated for several masses around the predicted mass of 6.27 GeV/$c^2$, and the results are shown in Table 3.4.

The model also includes a relativistic compensation parameter, $\kappa$, used to adjust the slope of the predicted pion form factor to match experiment. Since all the states considered here involve heavy quarks, $\kappa$ is simply set to one. Table 3.5 shows decay rates for $\kappa$ in the range of 0.7 to 1, where $\kappa = 0.7$ is the

Table 3.5: Decay rates as a function of the relativistic compensation factor, $\kappa$. These are all for the expected $B_c^{\pm}$ mass of 6.27 GeV/$c^2$. The decay rates are in units of $10^{10}s^{-1}$.

| $\kappa$ | $c\bar{c}$ | $c\bar{u}$ | $\bar{b}s$ | $\bar{b}d$ |
|---|---|---|---|---|
| 0.7 | 6.33 | 0.0359 | 6.73 | 0.371 |
| 0.8 | 7.90 | 0.0562 | 7.62 | 0.439 |
| 0.9 | 9.32 | 0.0823 | 8.32 | 0.496 |
| 1.0 | 10.58 | 0.1134 | 8.88 | 0.543 |

value used in the previous calculations of $B$ and $D$ decays.

## 3.4 A Comparison with the Spectator Model

The estimates for semileptonic $B_c$ decay using the spectator model by Du and Wang [18] vary considerably from the results obtained here. The comparison is complicated by the fact that the results in that paper are not presented in terms of the K-M matrix elements. Using their K-M values on this calculation allows a comparison and gives inclusive rates that are smaller than theirs, but exclusive rates that are comparable. For example, their total $c$-quark decay rate is four times the one calculated here (recall that the $c$-quark decay rate converges). However, surprisingly, the exclusive rates they provide agree up to the presented accuracy with the rates calculated here. This implies a disagreement in not only the total rate, but also a significant disagreement in certain branching ratios. For example, they predict that $c \to s$ decays into the two lowest $B_s$ states account for, at most, 29% of all decays of the $c$-quark (using constituent quark masses for both inclusive and exclusive decays). However one would expect from phase space considerations that the two lowest

$B_s$ states would dominate heavily. And in fact they do in the quark potential model calculations here, accounting for 93% of the $c$-quark decays.

## 3.5 Phase Space Surgery

Semileptonic decays of the $b$-quark can result in very high lepton ($e$ or $\mu$) energies, allowing cuts that select lepton energies above some value to distinguish such decays from the background, and furthermore, selecting out the low-lying final states which facilitates comparison with the results presented here. This is particularly suitable for the $b \to c$ (charmonium) decays since they account for about half of all the semileptonic decays and have lepton energies up to 2.42 GeV.

A cut selecting electron or muon energies above 2.00 GeV restricts the observations of $b \to c$ decays to those final states included in this calculation, and selects 3.4% of the $b$-quark decays. The 2.00 GeV cut is the highest lepton energy that can result from a decay into the $1^3\mathrm{D}_1$ $c\bar{c}$ state (the $\psi(3770)$) which is the lowest final state not included in the $b \to c$ calculation.

A lower cut, to include more of the $b \to c$ decays, can be selected by allowing the top 10% of the phase space of decays into the $\psi(3770)$. The top 10% of the phase space accounts for, typically, 3 to 4% of the decays, so this would not affect the accuracy of the prediction significantly, but would increase the number of measured events substantially. The cut in this case would be at 1.69 GeV, and would select 18.2% of the $b$-quark decays, or over five times as many events as the more stringent cut. The results for both kinds of cuts are

Table 3.6: Cuts on the lepton energy for decays to charmonia states. The decay rates are in units of $10^{10}\text{s}^{-1}|V_{cb}|^2$. The masses are in GeV/$c^2$ and the lepton energy cuts are in GeV.

| $B_c^\pm$ mass | 0% $1^3\text{D}_1$ cut | $\Gamma_{\text{cut}}$ | $\Gamma_{\text{cut}}/\Gamma_{b\to c}$ | 10% $1^3\text{D}_1$ cut | $\Gamma_{\text{cut}}$ | $\Gamma_{\text{cut}}/\Gamma_{b\to c}$ |
|---|---|---|---|---|---|---|
| 6.12 | 1.90 | 172. | 4.4% | 1.60 | 815. | 20.8% |
| 6.15 | 1.92 | 171. | 4.2% | 1.62 | 829. | 20.5% |
| 6.18 | 1.94 | 169. | 4.1% | 1.63 | 843. | 20.2% |
| 6.21 | 1.96 | 168. | 3.9% | 1.65 | 856. | 19.8% |
| 6.24 | 1.98 | 166. | 3.7% | 1.67 | 870. | 19.5% |
| 6.27 | 2.00 | 165. | 3.6% | 1.69 | 883. | 19.2% |
| 6.30 | 2.02 | 163. | 3.4% | 1.71 | 897. | 18.9% |
| 6.33 | 2.04 | 161. | 3.3% | 1.72 | 910. | 18.6% |
| 6.36 | 2.06 | 160. | 3.2% | 1.74 | 923. | 18.4% |
| 6.39 | 2.08 | 158. | 3.0% | 1.76 | 937. | 18.1% |
| 6.42 | 2.10 | 156. | 2.9% | 1.78 | 950. | 17.8% |

shown in Table 3.6 for different values of the $B_c$ mass. These endpoint cuts give rates that are much less sensitive to the $B_c^\pm$ mass than the total rates.

## 3.6 Polarization

Another prediction that can be extracted from the quark potential model is the polarization of the final meson, in particular, the spin 1 $1^3\text{S}_1$ $c\bar{c}$ state—the $J/\psi$. The $J/\psi$ decays into two electrons or two muons about 14% of the time [2], allowing not only an accurate measurement of the angular distribution of those decays, but also identification, having a very clear three-lepton signature. And, perhaps most importantly, the polarization prediction and observation is independent of $|V_{cb}|$.

Three lepton decays will be dominated by the $J/\psi$, the only other contribution coming from the $\psi(2\text{S})$, which is also too light to decay into $D\overline{D}$. The

$\psi(2S)$ decays into lepton pairs less often than the $J/\psi$ and also accounts for fewer of the $B_c^\pm$ decays. As a result, the $\psi(2S)$ would account for less than 2% of the three lepton events, and so would not contribute significantly to the measurement. Those few $\psi(2S)$ events are easily separated using the momenta of the last two leptons. It suffices then to calculate the polarization of the $J/\psi$.

Here we calculate the partial decay rates of $B_c^\pm$ into longitudinally and transversely polarized $J/\psi$'s, where the polarization is along the momentum vector of the $J/\psi$ in the $B_c$ rest frame. The partial decay rates are integrated over the entire range of energies of the lepton from the weak decay, since the three-lepton signature provides the identification. The polarization information is contained in the form factors $f$ and $a_+$, which arise from the axial part of the quark current between the $B_c$ and the $J/\psi$ [19, 20]. The results are shown in Table 3.7 as a function of the $B_c$ mass. It should be noted that recent measurements of polarization in $D \to \overline{K^*} e^+ \nu_e$, as well as the rate disagree with quark potential model predictions [21]. However, similar measurements and predictions of $\overline{B} \to D^* e^- \overline{\nu}_e$ do agree [21]. Since the latter is a $b \to c$ transition, it is of more consequence for the polarization of the $J/\psi$.

## 3.7 Summary

Detailed predictions of the quark potential model have been presented that allow its applicability to be tested for the heavy quark system, $B_c^\pm$. The model is expected to be quite good in this less relativistic regime. Measurements of the relative $b$-quark and $c$-quark decays distinguish the predictions of this model

Table 3.7: Polarization of $J/\psi$ along its momentum vector in the rest frame of the $B_c^{\pm}$, broken down into longitudinally ($J_z = 0$) and transversely ($J_z = \pm 1$) polarized states. The polarization measure shown is $(2\Gamma^{(L)} - \Gamma^{(T)})/(\Gamma^{(L)} + \Gamma^{(T)})$, which ranges from -1 to 2 and is zero for an unpolarized state. The decay rates are in units of $10^{10}\text{s}^{-1}|V_{cb}|^2$.

| $B_c^{\pm}$ mass | $\Gamma^{(L)}$ | $\Gamma^{(T)}$ | Polarization |
|---|---|---|---|
| 6.12 GeV/$c^2$ | 1238. | 1167. | 0.544 |
| 6.15 GeV/$c^2$ | 1269. | 1203. | 0.540 |
| 6.18 GeV/$c^2$ | 1299. | 1240. | 0.535 |
| 6.21 GeV/$c^2$ | 1329. | 1278. | 0.529 |
| 6.24 GeV/$c^2$ | 1359. | 1316. | 0.524 |
| 6.27 GeV/$c^2$ | 1389. | 1355. | 0.519 |
| 6.30 GeV/$c^2$ | 1419. | 1394. | 0.513 |
| 6.33 GeV/$c^2$ | 1449. | 1433. | 0.508 |
| 6.36 GeV/$c^2$ | 1479. | 1474. | 0.503 |
| 6.39 GeV/$c^2$ | 1509. | 1514. | 0.497 |
| 6.42 GeV/$c^2$ | 1538. | 1555. | 0.492 |

from those of the free quark or similar (more elaborate) spectator models. Despite the incompleteness of the $b$-quark decay calculation, cuts on lepton energy allow direct comparison of experiment with these results. Finally, the polarization of the $J/\psi$ final state, which is independent of $|V_{cb}|$, is predicted. Decays of the $J/\psi$ into two leptons should be easily distinguished from the background, with their angular distribution providing a clean measure of the polarization.

# Appendix A

# Mathematica Notebook

What follows is the Mathematica Notebook used to do the decay rate calculations. Version 1.2 (January 20, 1990) of the Mathematica kernel and version 1.2 of the Mathematica front end, both on the NeXT computer, were used.

# Semileptonic Charmed Beauty Meson Decay

All beauteous things for which we live
By laws of space and time decay.
But Oh, the very reason why
I clasp them, is because they die.

WILLIAM CORY 1823—1892

## Initialization

Isolate this notebook from other active notebooks in the Kernel:

```
If[!MemberQ[$ContextPath,"Global`"],
    EndPackage[]; EndPackage[]]
BeginPackage["isolate`"]
BeginPackage["bcbar`"]
If[!MemberQ[$ContextPath,"inte`"],
    PrependTo[$ContextPath,"inte`"]]
$ContextPath
```

```
{inte`, bcbar`, System`}
```

Kill all of our definitions to rerun after a change:

```
Remove["bcbar`*"]
```

## Parameters

### Constants

The width of the exclusive decay B->Xev is calculated where B is a p-r meson and X is a q-r meson. p, q, and r can take the values u, d, s, c, and b.

```
Gf := 1.16637 10^-5 GeV^-2
```

### Adjustments

kp is kappa---a fudge factor to get the pion form factor correct:

```
kp := 1
```

## ■ Effective quark masses for quark potential model

```
m[-(q_)] := m[q]

m[u] := 0.33 GeV

m[d] := m[u]

m[s] := 0.55 GeV

m[c] := 1.82 GeV

m[b] := 5.12 GeV
```

Quark charges (needed to determine the sign of the gamma term):

```
Q[-(q_)] := -Q[q]

         2
Q[u] := -
         3

          1
Q[d] := -(-)
          3

          1
Q[s] := -(-)
          3

         2
Q[c] := -
         3

          1
Q[b] := -(-)
          3
```

## ■ Meson masses

Note that the mass definitions obey isospin symmetry.

```
B := {p, r, h11S0}

X := {q, r, ho}

m[{-(f_), g_, h_}] := m[{f, g, h}]
```

```
m[{f_, -(g_), h_}] := m[{f, g, h}]

m[{f_, g_, h_}] := m[{g, f, h}] /; !OrderedQ[{f, g}]

m[{d, d, h_}] := m[{u, u, h}]

m[{d, s, h_}] := m[{s, u, h}]

m[{c, u, h_}] := m[{c, d, h}]

m[{b, d, h_}] := m[{b, u, h}]
```

□ **Isovector meson masses**

```
m[{d, u, h11S0}] := 0.14 GeV

m[{d, u, h13S1}] := 0.77 GeV

m[{d, u, h11P1}] := 1.23 GeV

m[{d, u, h13P2}] := 1.32 GeV

m[{d, u, h13P1}] := 1.28 GeV

m[{d, u, h13P0}] := 1.09 GeV

m[{d, u, h21S0}] := 1.3 GeV

m[{d, u, h23S1}] := 1.45 GeV
```

□ **Light Isoscalar meson masses**

Note: the mixing of eta and eta-prime are not properly treated yet, but they are not among the final states considered here anyway.

```
m[{u, u, h11S0}] := 0.55 GeV

m[{u, u, h13S1}] := 0.78 GeV

m[{u, u, h11P1}] := 1.19 GeV

m[{u, u, h13P2}] := 1.27 GeV

m[{u, u, h13P1}] := 1.28 GeV

m[{u, u, h13P0}] := 1.09 GeV

m[{u, u, h21S0}] := 1.44 GeV

m[{u, u, h23S1}] := 1.46 GeV
```

□ **Strange meson masses**

```
m[{s, u, h11S0}] := 0.49 GeV

m[{s, u, h13S1}] := 0.89 GeV

m[{s, u, h11P1}] := 1.27 GeV

m[{s, u, h13P2}] := 1.43 GeV

m[{s, u, h13P1}] := 1.41 GeV

m[{s, u, h13P0}] := 1.24 GeV

m[{s, u, h21S0}] := 1.45 GeV

m[{s, u, h23S1}] := 1.58 GeV
```

□ **Charmed meson masses**

```
m[{c, d, h11S0}] := 1.86 GeV

m[{c, d, h13S1}] := 2.01 GeV

m[{c, d, h11P1}] := 2.44 GeV

m[{c, d, h13P2}] := 2.5 GeV

m[{c, d, h13P1}] := 2.49 GeV

m[{c, d, h13P0}] := 2.4 GeV

m[{c, d, h21S0}] := 2.58 GeV

m[{c, d, h23S1}] := 2.64 GeV
```

The mass for the 13D1 state is from Godfrey and Isgur. It is to set an electron energy cut to exclude final states not included in the calculation (the first such state being the 13D1 state).

```
m[{c, d, h13D1}] := 2.82 GeV
```

□ **Charmonium meson masses**

```
m[{c, c, h11S0}] := 2.98 GeV

m[{c, c, h13S1}] := 3.1 GeV

m[{c, c, h11P1}] := 3.51 GeV
```

```
m[{c, c, h13P2}] := 3.56 GeV
```

```
m[{c, c, h13P1}] := 3.51 GeV
```

```
m[{c, c, h13P0}] := 3.42 GeV
```

```
m[{c, c, h21S0}] := 3.59 GeV
```

```
m[{c, c, h23S1}] := 3.69 GeV
```

The mass for the 13D1 state is from experiment. It is to set an electron energy cut to exclude final states not included in the calculation (the first such state being the 13D1 state).

```
m[{c, c, h13D1}] := 3.77 GeV
```

□ **Beauty meson masses**

```
m[{b, u, h11S0}] := 5.28 GeV
```

```
m[{b, u, h13S1}] := 5.33 GeV
```

The rest of the B meson masses are from Godfrey and Isgur:

```
m[{b, u, h11P1}] := 5.8 GeV
```

```
m[{b, u, h13P2}] := 5.8 GeV
```

```
m[{b, u, h13P1}] := 5.8 GeV
```

The 13P0 state is missing in Godfrey and Isgur, so this is a guess:

```
m[{b, u, h13P0}] := 5.75 GeV
```

```
m[{b, u, h21S0}] := 5.9 GeV
```

```
m[{b, u, h23S1}] := 5.93 GeV
```

□ **Strange Beauty meson masses**

The strange B meson messes are from Godfrey and Isgur:

```
m[{b, s, h11S0}] := 5.39 GeV
```

```
m[{b, s, h13S1}] := 5.45 GeV
```

```
m[{b, s, h11P1}] := 5.88 GeV
```

```
m[{b, s, h13P2}] := 5.88 GeV
```

```
m[{b, s, h13P1}] := 5.88 GeV
```

The 13P0 state is missing in Godfrey and Isgur, so this is a guess:

```
m[{b, s, h13P0}] := 5.83 GeV

m[{b, s, h21S0}] := 5.98 GeV

m[{b, s, h23S1}] := 6.01 GeV
```

### □ Charmed Beauty meson mass

The charmed B meson mass is from Godfrey and Isgur. The calculation is quite sensitive to this mass.

```
m[{b, c, h11S0}] := 6.27 GeV
```

## ■ Meson Betas

```
bt[{-(f_), g_, h_}] := bt[{f, g, h}]

bt[{f_, -(g_), h_}] := bt[{f, g, h}]

bt[{f_, g_, h11S0}] := bt[{f, g, S}]

bt[{f_, g_, h13S1}] := bt[{f, g, S}]

bt[{f_, g_, h11P1}] := bt[{f, g, P}]

bt[{f_, g_, h13P2}] := bt[{f, g, P}]

bt[{f_, g_, h13P1}] := bt[{f, g, P}]

bt[{f_, g_, h13P0}] := bt[{f, g, P}]

bt[{f_, g_, h21S0}] := bt[{f, g, S}]

bt[{f_, g_, h23S1}] := bt[{f, g, S}]

bt[{f_, d, h_}] := bt[{f, u, h}]

bt[{d, g_, h_}] := bt[{u, g, h}]

bt[{f_, g_, h_}] := bt[{g, f, h}] /; !OrderedQ[{f, g}]

bt[{u, u, S}] := 0.31 GeV

bt[{s, u, S}] := 0.34 GeV

bt[{c, u, S}] := 0.39 GeV
```

```
bt[{b, u, S}] := 0.41 GeV
```

```
bt[{u, u, P}] := 0.27 GeV
```

```
bt[{s, u, P}] := 0.3 GeV
```

```
bt[{c, u, P}] := 0.34 GeV
```

Additional Betas for Bc decay (cs, bs are with strong alpha=0.5, cc and bc use alpha= 0.4):

```
bt[{b, u, P}] := 0.35 GeV
```

```
bt[{c, s, S}] := 0.47 GeV
```

```
bt[{b, s, S}] := 0.51 GeV
```

```
bt[{c, c, S}] := 0.65 GeV
```

```
bt[{b, c, S}] := 0.81 GeV
```

```
bt[{c, s, P}] := 0.39 GeV
```

```
bt[{b, s, P}] := 0.41 GeV
```

```
bt[{c, c, P}] := 0.52 GeV
```

## ■ Kobayashi-Maskawa matrix

```
V[-(q_), p_] := V[q, p]
```

```
V[q_, -(p_)] := V[q, p]
```

```
V[q_, p_] := V[p, q] /; !OrderedQ[{q, p}]
```

These are actually the absolute values of the matrix elements. They are only approximate (especially Vub and Vcb), and the t quark row is left out:

```
V[d, u] := 0.9755
```

```
V[s, u] := 0.2199
```

```
V[b, u] := 0.0053
```

```
V[c, d] := 0.2197
```

```
V[c, s] := 0.9744
```

```
V[b, c] := 0.048
```

# Equations

## Width

```
                        2
width := Abs[V[q, p]]  widthhat

widthhat :=

     2      5   al[ho] y
  (Gf  m[B]  ((-------- +
                      2
                 m[B]


                                        2
                                    m[X]             2
        2 btpp[ho] (2 x (1 - ----- + y) - 4 x  - y)) -
                                        2
                                    m[B]


                                             2
                                         m[X]
      (Q[q] - Q[p]) gm[ho] y (1 - ----- - 4 x + y))) /
                                             2
                                         m[B]


          3
   (32 N[Pi] )

widthhatL :=


                                                2
   2      5                                 m[X]              2
  Gf  m[B]  2 btppL[ho] (2 x (1 - ----- + y) - 4 x  - y)
                                                2
                                            m[B]
  ------------------------------------------------------
                                  3
                           32 N[Pi]
```

```
widthhatT :=

     2      5   al[ho] y
  (Gf  m[B]   ((-------- +
                      2
                  m[B]

                                              2
                                          m[X]          2
          2 btppT[ho] (2 x (1 - ----- + y) - 4 x  - y)) -
                                              2
                                          m[B]

                                                  2
                                              m[X]
        (Q[q] - Q[p]) gm[ho] y (1 - ----- - 4 x + y))) /
                                                  2
                                              m[B]

            3
   (32 N[Pi] )
```

## ■ Common Functions

```
              2
t := m[B]  y

mh[{f_, g_, h_}] := m[f] + m[g]

           mh[X] 1/2  bt[B] bt[X] n/2
F[n_] := (-----)     (-----------)
           mh[B]                  2
                          bt[BX]

                  2
              m[r]  (tm - t)
   Exp[-(--------------------------)]
                              2       2
        4 mh[B] mh[X] kp  bt[BX]
```

```
                          2         2
                     bt[B]   + bt[X]
bt[BX] := Sqrt[---------------]
                            2

                         2
tm := (m[B] - m[X])

           1      1   -1
mup := (---- + ----)
         m[q]   m[p]

           1      1   -1
mum := (---- - ----)
         m[q]   m[p]

                      2                2 2
               (m[B]   (1 - y) + m[X] )         2
px := Sqrt[------------------------ - m[X] ]
                                 2
                            4 m[B]
```

## ■ Harmonic Oscillator State n=1, L=0, and spin S=0

```
                         2
al[h11S0] := 0 GeV

                         2
btpp[h11S0] := fpl

gm[h11S0] := 0

                                                      2
                        m[p]       m[p] m[q] m[r] bt[B]
fpl := F[3] ((1 + -----) - -----------------------)
                        2 mum                                2
                                   4 mup mum mh[X] bt[BX]
```

## ■ Harmonic Oscillator State n=1, L=0, and spin S=1

```
                      2          2  2   2
al[h13S1] := f  + 4 m[B]  g  px
```

For this state, we define, in addition, the longitudinal and transverse polarizations:

**btppL[h13S1] :=**

$$\frac{\left(\frac{m[B]^2 (1 - y)}{m[X]^2} - 1\right) f\, ap}{2} + \frac{m[B]^2\, px^2\, ap^2}{m[X]^2} + \frac{\left(\frac{m[B]\,(1 - y)}{m[X]} - \frac{m[X]}{m[B]}\right)^2 f^2}{16\, px^2}$$

**btppT[h13S1] :=**

$$\frac{f^2}{4\, m[X]^2} - m[B]^2\, g^2\, y - \frac{\left(\frac{m[B]\,(1 - y)}{m[X]} - \frac{m[X]}{m[B]}\right)^2 f^2}{16\, px^2}$$

btpp is the sum of btppL and btppT:

```
btpp[h13S1] :=

                                2
                           m[B]   (1 - y)
                          (------------- - 1) f ap
     2                                  2
    f           2  2              m[X]
  ------- - m[B]   g   y + ------------------------ +
        2                                  2
  4 m[X]

        2   2   2
   m[B]   px   ap
   -------------
            2
       m[X]
```

```
gm[h13S1] := 2 g f
```

```
f := 2 mh[B] F[3]
```

```
                                  2
            1        1 m[r] bt[B]
     F[3] (---- - -------------------)
           m[q]                    2
                  2 mum mh[X] bt[BX]
g := ---------------------------------
                     2
```

```
                                 2         2
                    m[r] (bt[B]   - bt[X] )
ap := -((F[3] ((1 + ----------------------) -
                                 2         2
                    m[p] (bt[B]   + bt[X] )

                2      4
            m[r]   bt[X]
         --------------------)) / (2 mh[X]))
                           4
         4 mum mh[B] bt[BX]
```

## ■ Harmonic Oscillator State n=1, L=1, and spin S=1, J=2

$$\mathrm{al[h13P2]} := \frac{m[B]^2\, px^2\, (k^2 + 4\, m[B]^2\, px^2\, h^2)}{2\, m[X]^2}$$

btpp[h13P2] :=

$$-\left(\frac{y\, m[B]^4\, px^2\, h^2}{2\, m[X]^2}\right) + \frac{m[B]^2 \left(y + \frac{4\, px^2}{m[X]^2}\right) k^2}{24\, m[X]^2} +$$

$$\frac{2 \left(\frac{m[B]^2\, px^2}{m[X]^2}\right)^2 bp^2}{3} +$$

$$\frac{m[B]^2\, px^2 \left(\frac{2\, m[B]^2\, (1 - y)}{m[X]^2} - 1\right) k\, bp}{m[X]^2\, 3}$$

$$\mathrm{gm[h13P2]} := \frac{m[B]^2\, px^2\, k\, h}{m[X]^2}$$

```
                                        2
                 1         m[r] bt[B]
     F[5] m[r] (---- - -------------------)
                m[q]                   2
                       2 mh[X] mum bt[BX]
h := --------------------------------------
             2 Sqrt[2] mh[B] bt[B]

     Sqrt[2] F[5] m[r]
k := -----------------
           bt[B]

                                           2
                            m[r] m[p] bt[X]
bp := -((F[5] m[r] (1 - ------------------- +
                                             2
                         2 mup mh[B] bt[BX]

                                                2
                         2        m[r] bt[X]
         m[r] m[p] bt[X]   (1 - ---------------)
                                                  2
                                  2 mh[B] bt[BX]
         --------------------------------------)) /
                                         2
                    4 mh[B] mum bt[BX]

     (2 Sqrt[2] mh[X] m[p] bt[B]))
```

## ■ Harmonic Oscillator State n=1, L=1, and spin S=1, J=1

```
                 2          2   2   2
al[h13P1] := 1   + 4 m[B]   px  qq
```

```
btpp[h13P1] :=

                                        2
                                   m[B]   (1 - y)
                                  (------------- - 1) l cp
      2                                     2
     1             2       2           m[X]
   ------- - m[B]   y qq   + ----------------------- +
         2                                    2
   4 m[X]

        2    2    2
    m[B]   px   cp
    -------------
             2
        m[X]

gm[h13P1] := 2 qq l

         F[5] m[r]
qq := -------------
       2 mh[X] bt[B]

l := -(F[5] mh[B] bt[B]

                                                           2
                              1               m[r] bt[B]
             m[r] (tm - t) (---- - -------------------)
                            m[q]                         2
     1                                 2 mum mh[X] bt[BX]
    (--- + -----------------------------------------------))
    mum                                    2       2
                               2 mh[B] kp   bt[B]

                                                2
                              m[r] m[q] bt[B]
      F[5] m[r] m[p] (1 - -------------------)
                                                  2
                            2 mh[X] mum bt[BX]
cp := -------------------------------------------
                     4 mh[B] bt[B] mum
```

## ■ Harmonic Oscillator State n=1, L=1, and spin S=1, J=0

```
                       2
al[h13P0] := 0 GeV

                      2
btpp[h13P0] := up

gm[h13P0] := 0

         F[5] m[r] m[q] m[p]
up := -----------------------
       Sqrt[6] bt[B] mh[X] mum
```

## ■ Harmonic Oscillator State n=1, L=1, and spin S=0

```
                  2          2   2  2
al[h11P1] := rr   + 4 m[B]   px   v

btpp[h11P1] :=

                                   2
                              m[B]   (1 - y)
                             (------------- - 1) rr sp
    2                                  2
   rr          2     2               m[X]
 ------- - m[B]   y v   + ------------------------- +
      2                                 2
 4 m[X]

     2   2   2
 m[B]   px   sp
 -------------
         2
     m[X]
```

```
gm[h11P1] := 2 rr v

          F[5] mh[B] bt[B]
v := -------------------------
     4 Sqrt[2] m[p] m[q] mh[X]

       F[5] mh[B] bt[B]
rr := ----------------
         Sqrt[2] mup

                                                          2
                        m[p]      m[p] m[q] m[r] bt[B]
      F[5] m[r] ((1 + -----) - -----------------------)
                        2 mum                               2
                                4 mup mum mh[X] bt[BX]
sp := --------------------------------------------------
                       Sqrt[2] bt[B] mh[B]
```

## ■ Harmonic Oscillator State n=2, L=0, and spin S=0

```
                    2
al[h21S0] := 0 GeV

                    2
btpp[h21S0] := fpp

gm[h21S0] := 0
```

```
               3 1/2
fpp := (F[3] (-)     m[p]
               8

             2         2
      bt[B]   - bt[X]
     (-------------- +
             2         2
      bt[B]   + bt[X]

                       2          2           2
       m[q] m[r] bt[B]   (7 bt[X]   - 3 bt[B] )
       -------------------------------------- +
                                2          2
           3 mum mh[X] bt[BX]   4 bt[BX]

                                                        2
           2      2                   m[q] m[r] bt[B]
       m[r]   bt[X]   (tm - t) (1 - -------------------)
                                                          2
                                    2 mum mh[X] bt[BX]
       ----------------------------------------------))\
                                    2   2        2
              6 mh[X] mh[B] bt[BX]   kp   bt[BX]

     / mup
```

## ■ Harmonic Oscillator State n=2, L=0, and spin S=1

$$\mathtt{al[h23S1]} := fp^2 + 4\, m[B]^2\, px^2\, gp^2$$

$$\mathtt{btpp[h23S1]} :=$$

$$\frac{fp^2}{4\, m[X]^2} - m[B]^2\, y\, gp^2 + \frac{\left(\frac{m[B]^2 (1 - y)}{m[X]^2} - 1\right) fp\ app}{2} + \frac{m[B]^2\, px^2\, app^2}{m[X]^2}$$

$$\mathtt{gm[h23S1]} := 2\, gp\, fp$$

```
fp := Sqrt[6] F[3] mh[B]

         2        2             2     2
    bt[B]  - bt[X]          m[r]  bt[X]  (tm - t)
   (--------------- + ---------------------------------)
         2        2                          2  2     2
    bt[B]  + bt[X]    6 mh[X] mh[B] bt[BX]  kp  bt[BX]

                           2        2
      3 1/2          bt[B]  - bt[X]
gp := (-)     F[3] ((--------------- +
      8                    2        2
                     bt[B]  + bt[X]

                 2      2
            m[r]  bt[X]  (tm - t)
     ---------------------------------)
                              2   2      2
     6 mh[X] mh[B] bt[BX]  kp  bt[BX]

                          2                 2      2
      1          m[r] bt[B]          m[r] bt[B]  bt[X]
    (---- - -------------------) + -------------------)
     m[q]                      2                     4
            2 mum mh[X] bt[BX]     3 mum mh[X] bt[BX]
```

```
app := (F[5] ((

                                       2            4
                    2            m[r]   m[p]  bt[X]
     3 mh[B] bt[BX]   (1 - --------------------)
                                         2            4
                                4 mh[B]   mum bt[BX]
     ------------------------------------------- -
                   2 m[p] bt[B] bt[X]

                                                  T
                    5 m[r] bt[B] bt[X] (1 + --)
    3 m[r] bt[X]                                  10
    ------------ + ---------------------------) -
    2 m[p] bt[B]                           2
                             2 m[p] bt[BX]

                       T
   3 mh[B] bt[B] (1 + -)
                       6
   -------------------- +
      2 m[p] bt[X]

        2              4       T
  7 m[r]  bt[B] bt[X]   (1 + --)
                               14
  ----------------------------)) / (Sqrt[6] mh[X])
                               4
    8 mh[B] mum bt[X] bt[BX]

        2       2
    m[r]  bt[X]  (tm - t)
T := ----------------------------
                        2   2        2
    mh[X] mh[B] bt[B]  kp   bt[BX]
```

## Integrator

This integrator only handles indefinite integrals of the form: E^(p x)P[x], where P[x] is a polynomial. It performs the integration in a way that leaves the E^(p x) on the outside.

```
BeginPackage["inte`"]
```

```
inte`
```

```
inte::usage =
    "inte[f, p, x] is equivalent to:
    E^(-p x)Integrate[E^(p x) f, x]
    where f only contains terms like E^(q x) x^n where n is
    a non-negative integer."
```

```
inte[f, p, x] is equivalent to:  E^(-p x)Integrate[E^(p\

  x) f, x]  where f only contains terms like E^(q x) x^n\

  where n is  a non-negative integer.
```

```
Begin["`private`"]
```

```
inte`private`
```

```
inte[c_ f_,p_,x_] := c inte[f,p,x] /; FreeQ[c,x]
inte[f_+g_,p_,x_] := inte[f,p,x]+inte[g,p,x]
inte[x_^n_.(f_+g_),p_,x_] := inte[x^n f+x^n g,p,x] /; !FreeQ[f,x]
inte[c_,0,x_] := c x /; FreeQ[c,x]
inte[c_,p_,x_] := c/p /; FreeQ[c,x]
inte[E^(q_. x_)f_.,p_,x_] := E^(q x)inte[f,p+q,x] /; FreeQ[q,x]
inte[x_^n_.,0,x_] := x^(n+1)/(n+1) /; n != -1
inte[x_^n_.,p_,x_] := x^n/p-n/p inte[x^(n-1),p,x] /;
    IntegerQ[n] && n > 0
```

```
End[]
```

```
inte`private`
```

```
EndPackage[]
```

## ▨ Calculator

Given a pr meson, compute gammahat, yupper, and xupper for all accessible and defined states qr, where p, q, and r can take the values u, d, s, c, or b. The pr is assumed to be in the 11S0 state, and qr is tried in all 1S, 1P, and 2S states. It is assumed that r is not the antiparticle of p. Also, a list of states done is kept in 'process' and the mass of the pr meson is saved in mB. The full definitions of these five names (gammahat, yupper, xupper, process, and mB) are saved in a file named "prdecay.m" where pr is replaced by the sorted, unsigned values of p and r. For example, domeson[c, -b] would save its results in a file bcdecay.m.

```
domeson[p_, r_] := Block[ {qlist, i},
    process[] = {} ;
    (* W couplings for p that reduce the mass *)
    qlist = Switch[p,
            b, {c, u}, -b, {-c, -u},
            c, {s, d}, -c, {-s, -d},
            s, {u}, -s, {-u},
            u, {d}, -u, {-d},
            d, {u}, -d, {-u},
            _, {}] ;
    Do[
        AppendTo[process[], {p, qlist[[i]], r}] ;
        process[{p, qlist[[i]], r}] = {} ;
        dodecay[p, qlist[[i]], r] ;
        , {i, Length[qlist]}
    ] ;
    (* W couplings for r that reduce the mass *)
    qlist = Switch[r,
            b, {c, u}, -b, {-c, -u},
            c, {s, d}, -c, {-s, -d},
            s, {u}, -s, {-u},
            u, {d}, -u, {-d},
            d, {u}, -d, {-u},
            _, {}] ;
    Do[
        AppendTo[process[], {r, qlist[[i]], p}] ;
        process[{r, qlist[[i]], p}] = {} ;
        dodecay[r, qlist[[i]], p] ;
        , {i, Length[qlist]}
    ] ;
    mB = m[{p, r, h11S0}] ;
    Put[FullDefinition[process, gammahat, gammahatL, gammahatLI,
        gammahatLX, gammahatT, gammahatTI, gammahatTX,
        gammahatxlb, gammahatyub, yupper, xupper, mB],
        Apply[StringJoin, Append[Map[ToString,
        Sort[{p, r} /. -x_ :> x]],
        "decay.m"]]] ;
]
```

```
dodecay[p_, q_, r_] := Block[ {ho, i},
    (* step through all 1S, 1P, and 2S states *)
    Do[
        ho = {h11S0, h13S1, h11P1, h13P2, h13P1,
            h13P0, h21S0, h23S1}[[i]] ;
        If[ m[{p, r, h11S0}] >
            m[{q, r, ho}] /. GeV -> 1
        (* then *) ,
            AppendTo[process[{p, q, r}],
                {p, q, r, ho}] ;
            doprocess[{p, q, r, ho}] ;
            Print["computed ", p, r, "-->", q, r,
                " in ", ho, " state"] ;
        (* else *) ,
            Null
        ] ;
       , {i, 8}
    ] ;
 ]
```

Collect cerrtain terms in expressions:

```
colGeV[f_] := f //. (a_. GeV^n_.+b_. GeV^n_.) :> (a+b) GeV^n
```

```
colexp[f_] := f //. {E^(a_+x_.) :> N[E^a]E^x /; NumberQ[a],
    E^x_ g_.+E^x_ h_. :> E^x(g+h)}
```

nint is a numerical integrator to help bypass the odd "AccuracyGoal" behavior of NIntegrate. nint uses gauss5 and NIntegrate to perform NIntegrate with a magnitude independent behavior. This only works if sampling five of the function points can give an order of magnitude estimate of the integral. Options given to nint will be passed on to NIntegrate. gauss5 does a Gaussian quadrature integration by sampling five points selected using Legendre polynomials. gausscoeff computes those sample points and associated coefficients just once when gauss5 is defined. gauss5 does not sample the endpoints of the integration, but does sample the midpoint.

```
gausscoeff[n_] := Block[{r,j,x},
    r = (#[[1,2]])& /@ Solve[LegendreP[n,x]==0,x] ;
    {r,2(Expand[Product[If[j!=#, (x-r[[j]])/(r[[#]]-r[[j]]),1],
    {j,n}]] /. {x^j_?EvenQ->1/(j+1),x^j_?OddQ->0} /. x->0)& /@
    Range[n]}]
```

```
`gauss`gscf5 = gausscoeff[5] // N ;
gauss5[f_,{x_,a_,b_}] := (b-a)/2 Plus @@
    (((f /. x->#)& /@ ((b+a+(b-a)`gauss`gscf5[[1]])/2)) *
            `gauss`gscf5[[2]])

nint[f_,{x_,a_,b_},o___] := (# NIntegrate[f/#,{x,a,b},o])&[
    gauss5[f,{x,a,b}]]

nint[f_,{x_,a_,b_,c_},o___] := (# NIntegrate[f/#,{x,a,b,c},o])&[
    gauss5[f,{x,a,b}]+gauss5[f,{x,b,c}]]
```

Given the quark constituents (p, q, and r) and the harmonic oscillator state of X, this program calculates the differential cross section in terms of x and y, saves that, integrates over y and saves that, and finally integrates over x and saves that. The results are with the K-M matrix element (squared) left off and the dimension (GeV) removed. The upper limits for x and y are also computed and saved. Also, if the 13S1 state is being done, compute the longitudinal and transverse polarizations. The final numerical integration in those cases is done just short of the upper limit to avoid the 0/0 condition there. The accuracy of the result is not affected.

```
doprocess[{parg_, qarg_, rarg_, harg_}] :=
    Block[ {p, q, r, ho, x, y, tmp},
        p = parg ;
        q = qarg ;
        r = rarg ;
        ho = harg ;
        yupper[{p, q, r, ho}][x_] =
            2x(1-(m[X]/m[B])^2/(1-2x)) ;
        yupper[{p, q, r, ho}][] =
            (1 - m[X]/m[B])^2 ;
        xupper[{p, q, r, ho}] =
            (1-(m[X]/m[B])^2)/2 ;
        tmp[x_, y_] = widthhat/GeV //.
            a_^b_ :> N[a^b] /; NumberQ[a] && NumberQ[b] //
            Simplify // colGeV // colexp ;
        gammahat[{p, q, r, ho}][x_, y_] =
            If[Release[x > 0 && y > 0 &&
                x < xupper[{p, q, r, ho}] &&
                y < yupper[{p, q, r, ho}][x]],
                Release[tmp[x, y]], 0] ;
        tmp[x_,y_] =
            inte[gammahat[{p, q, r, ho}][x, y][[2]] //
```

```
        ExpandAll, 0, y] ;
tmp[x_] = tmp[x, yupper[{p, q, r, ho}][x]]-tmp[x, 0] //
    Simplify // colexp;
gammahat[{p, q, r, ho}][x_] =
    If[Release[x > 0 && x < xupper[{p, q, r, ho}]],
        Release[tmp[x]], 0] ;
gammahat[{p, q, r, ho}][] = nint[
    gammahat[{p, q, r, ho}][x][[2]],
    {x, 0, xupper[{p, q, r, ho}]}] ;
If[ho==h13S1,
    (* Lower bound of x for polarization integration of
       h13S1 final state is upper bound of x for h23S1
       state. *)
    gammahatxlb[{p, q, r, ho}] =
        (1-(m[{q, r, h23S1}]/m[B])^2)/2 ;
    (* Upper bound of y for the integration is (almost)
       peak y if x lower bound is before y peak, or is
       determined by x lower bound if it is after the y
       peak. *)
    gammahatyub[{p, q, r, ho}] =
        If[gammahatxlb[{p, q, r, ho}] > (1-m[X]/m[B])/2,
            yupper[{p, q, r, ho}][
                gammahatxlb[{p, q, r, ho}]],
            0.999999 yupper[{p, q, r, ho}][]] ;
    (* Do longitudinal polarization *)
    tmp[x_, y_] = widthhatL/GeV //.
        a_^b_ :> N[a^b] /; NumberQ[a] && NumberQ[b] //
        Simplify // colGeV // colexp ;
    gammahatL[{p, q, r, ho}][x_, y_] =
        If[Release[x > 0 && y > 0 &&
            x < xupper[{p, q, r, ho}] &&
            y < yupper[{p, q, r, ho}][x]],
            Release[tmp[x, y]], 0] ;
    gammahatLI[{p, q, r, ho}][x_, y_] =
        inte[gammahatL[{p, q, r, ho}][x, y][[2]],
            0, x] ;
    gammahatLX[proc_][xlb_, y_] :=
        ((gammahatLI[proc][(#1+#3)/4,#2]-
                gammahatLI[proc][Max[xlb,(#1-#3)/4],#2]
            )&[2#1+#2,#2,Sqrt[(2#1+#2)^2-4#2]]
        )&[xupper[proc],y] ;
```

```
        gammahatL[{p, q, r, ho}][] =
            nint[gammahatLX[{p, q, r, ho}][
                gammahatxlb[{p, q, r, ho}],y],
                {y,0,yupper[{p, q, r, ho}][
                    gammahatxlb[{p, q, r, ho}]],#}]&[
                gammahatyub[{p, q, r, ho}]] ;
        (* Do transverse polarization *)
        tmp[x_, y_] = widthhatT/GeV //.
            a_^b_ :> N[a^b] /; NumberQ[a] && NumberQ[b] //
            Simplify // colGeV // colexp ;
        gammahatT[{p, q, r, ho}][x_, y_] =
            If[Release[x > 0 && y > 0 &&
                x < xupper[{p, q, r, ho}] &&
                y < yupper[{p, q, r, ho}][x]],
                Release[tmp[x, y]], 0] ;
        gammahatTI[{p, q, r, ho}][x_, y_] =
            inte[gammahatT[{p, q, r, ho}][x, y][[2]],
                0, x] ;
        gammahatTX[proc_][xlb_, y_] :=
            ((gammahatTI[proc][(#1+#3)/4,#2]-
                    gammahatTI[proc][Max[xlb,(#1-#3)/4],#2]
                )&[2#1+#2,#2,Sqrt[(2#1+#2)^2-4#2]]
            )&[xupper[proc],y] ;
        gammahatT[{p, q, r, ho}][] =
            nint[gammahatTX[{p, q, r, ho}][
                gammahatxlb[{p, q, r, ho}],y],
                {y,0,yupper[{p, q, r, ho}][
                    gammahatxlb[{p, q, r, ho}]],#}]&[
                gammahatyub[{p, q, r, ho}]] ;
    ]
]
```

## B to Xu Decay (test case)

This is a test case used to check against the same calculation done in Isgur, Scora, Grinstein, and Wise. The data shown here is the same as that in the graph and caption of Fig. 6 of that paper.

```
domeson[b, -d]
```

```
computed b-d-->c-d in h11S0 state
computed b-d-->c-d in h13S1 state
computed b-d-->c-d in h11P1 state
computed b-d-->c-d in h13P2 state
computed b-d-->c-d in h13P1 state
computed b-d-->c-d in h13P0 state
computed b-d-->c-d in h21S0 state
computed b-d-->c-d in h23S1 state
computed b-d-->u-d in h11S0 state
computed b-d-->u-d in h13S1 state
computed b-d-->u-d in h11P1 state
computed b-d-->u-d in h13P2 state
computed b-d-->u-d in h13P1 state
computed b-d-->u-d in h13P0 state
computed b-d-->u-d in h21S0 state
computed b-d-->u-d in h23S1 state
```

Partial rates of exclusive B --> Xu channels in units of 10^14 s^-1 |Vub|^2:

```
{#[[4]],gammahat[#][] 1.51926689 10^10}& /@
    process[process[][[2]]]
```

```
{{h11S0, 0.0209767724085388},
  {h13S1, 0.0827077099934149},
  {h11P1, 0.05922769633869074},
  {h13P2, 0.006524836229448925},
  {h13P1, 0.092424852466141},
  {h13P0, 0.006641548603414723},
  {h21S0, 0.1097502660051351},
  {h23S1, 0.05275278720997787}}
```

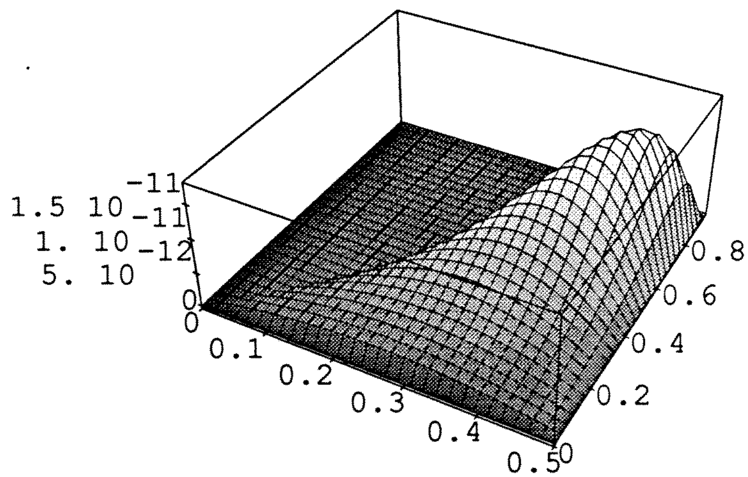
Total rate for B to Xu in units of 10^14 s^-1 |Vub|^2:

```
Apply[Plus,Map[(gammahat[#][] 1.51926689 10^10)&,
    process[process[][[2]]]]]
```

```
0.4310064692547619
```

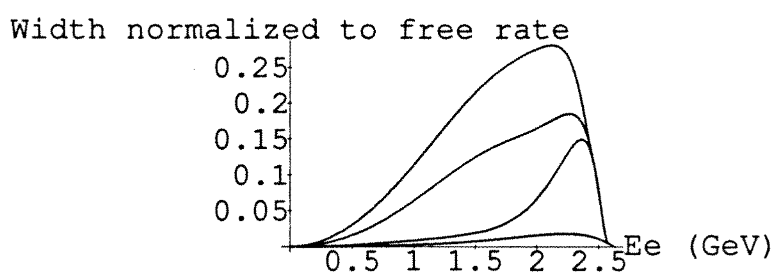
Plot of B --> pi partial width in terms of x and y:

```
proc = process[process[][[2]]][[1]]
Plot3D[gammahat[proc][x, y],
    Release[{x, 0, xupper[proc]}],
    Release[{y, 0, yupper[proc][]}],
    PlotPoints -> 30,
    PlotRange -> All]
```




Plot of contributions of pi, pho, 1P, and 1S states to B --> Xu:

```
Plot[
    Release[Map[#/(7.76 10^-11 mB/GeV)&,
        {gammahat[process[process[][[2]]][[1]]]
            [Ee/mB GeV],
        Apply[Plus, Map[Apply[#,{Ee/mB GeV}]&,
            Map[gammahat,
            process[process[][[2]]][[Range[2]]]]]],
        Apply[Plus, Map[Apply[#,{Ee/mB GeV}]&,
            Map[gammahat,
            process[process[][[2]]][[Range[6]]]]]],
        Apply[Plus, Map[Apply[#,{Ee/mB GeV}]&,
            Map[gammahat,
            process[process[][[2]]]]]]
        }]],
    Release[{Ee, 0, mB/GeV Apply[Max, Map[xupper,
        process[process[][[2]]]]]}],
    AxesLabel ->
        {"Ee (GeV)",
         "Width normalized to free rate"}]
```



## Bc Decay

```
domeson[-b, c]
```

```
computed -bc-->-cc in h11S0 state
computed -bc-->-cc in h13S1 state
computed -bc-->-cc in h11P1 state
computed -bc-->-cc in h13P2 state
computed -bc-->-cc in h13P1 state
computed -bc-->-cc in h13P0 state
computed -bc-->-cc in h21S0 state
computed -bc-->-cc in h23S1 state
computed -bc-->-uc in h11S0 state
computed -bc-->-uc in h13S1 state
computed -bc-->-uc in h11P1 state
computed -bc-->-uc in h13P2 state
computed -bc-->-uc in h13P1 state
computed -bc-->-uc in h13P0 state
computed -bc-->-uc in h21S0 state
computed -bc-->-uc in h23S1 state
computed c-b-->s-b in h11S0 state
computed c-b-->s-b in h13S1 state
computed c-b-->s-b in h11P1 state
computed c-b-->s-b in h13P2 state
computed c-b-->s-b in h13P1 state
computed c-b-->s-b in h13P0 state
computed c-b-->s-b in h21S0 state
computed c-b-->s-b in h23S1 state
computed c-b-->d-b in h11S0 state
computed c-b-->d-b in h13S1 state
computed c-b-->d-b in h11P1 state
computed c-b-->d-b in h13P2 state
computed c-b-->d-b in h13P1 state
computed c-b-->d-b in h13P0 state
computed c-b-->d-b in h21S0 state
computed c-b-->d-b in h23S1 state
```

Widths for the computed channels in 10^10 s^-1 |Vqp|^2:

```
Map[{#,gammahat[#][] 1.51926689 10^14}&,
    Map[process[#]&, process[]], {2}]
```

```
{{{{-b, -c, c, h11S0}, 750.2165550883402},
  {{-b, -c, c, h13S1}, 2743.591610691613},
  {{-b, -c, c, h11P1}, 317.304687133902},
  {{-b, -c, c, h13P2}, 180.5681663933378},
  {{-b, -c, c, h13P1}, 107.6697190342695},
  {{-b, -c, c, h13P0}, 94.3223608868918},
  {{-b, -c, c, h21S0}, 169.659797803702},
  {{-b, -c, c, h23S1}, 229.7948979013836}},
 {{{-b, -u, c, h11S0}, 233.6114690762324},
  {{-b, -u, c, h13S1}, 929.497811328875},
  {{-b, -u, c, h11P1}, 574.1864680456154},
  {{-b, -u, c, h13P2}, 96.1744952794525},
  {{-b, -u, c, h13P1}, 554.2587423626268},
  {{-b, -u, c, h13P0}, 48.90942483221627},
  {{-b, -u, c, h21S0}, 793.8788772362192},
  {{-b, -u, c, h23S1}, 807.136017205634}},
 {{{c, s, -b, h11S0}, 2.652769932793209},
  {{c, s, -b, h13S1}, 6.578714323321675},
  {{c, s, -b, h11P1}, 0.05525677096849717},
  {{c, s, -b, h13P2}, 0.001910231873915464},
  {{c, s, -b, h13P1}, 0.02445703462521035},
  {{c, s, -b, h13P0}, 0.01365511643558893},
  {{c, s, -b, h21S0}, 0.01443290580134082},
  {{c, s, -b, h23S1}, 0.00850156610975891}},
 {{{c, d, -b, h11S0}, 3.042393492466348},
  {{c, d, -b, h13S1}, 7.769855099674041},
  {{c, d, -b, h11P1}, 0.1723243590653841},
  {{c, d, -b, h13P2}, 0.003920407298234469},
  {{c, d, -b, h13P1}, 0.1314659492575735},
  {{c, d, -b, h13P0}, 0.02508822421610784},
  {{c, d, -b, h21S0}, 0.06801593542727014},
  {{c, d, -b, h23S1}, 0.040139554172089}}}
```

Total widths for the 1S, 1P, and 2S channels in 10^10 s^-1 |Vqp|^2:

```
Print[Take[#,2]& /@ process[]]
Apply[Plus,
    Map[(gammahat[#][] 1.51926689 10^14)&,
        Map[process[#]&, process[]], {2}],
    {1}]
```

```
{{-b, -c}, {-b, -u}, {c, s}, {c, d}}
```

```
{4593.13, 4037.65, 9.3497, 11.2532}
```

Total Widths with approximate K-M matrix values in 10^10 s^-1:

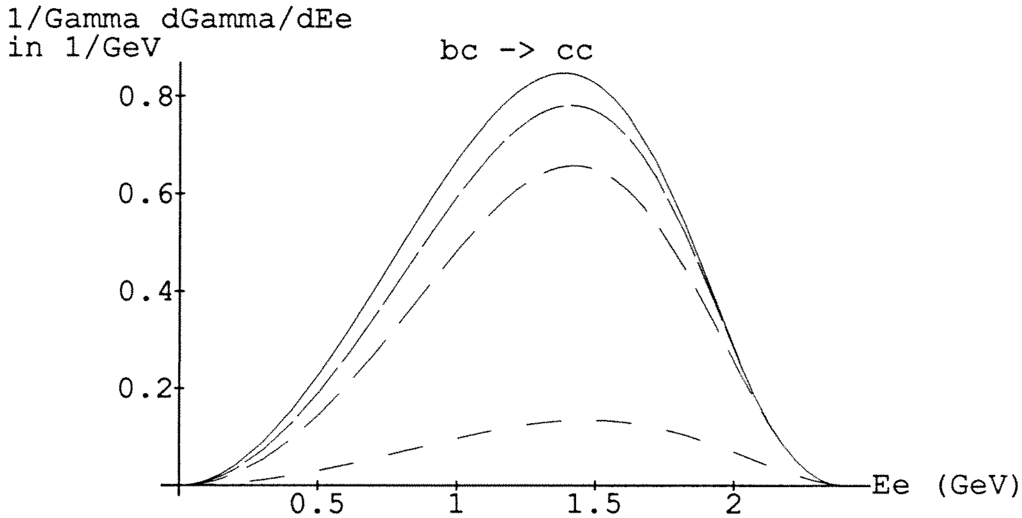
```
Print[Take[#,2]& /@ process[]]
Apply[Plus,
    Map[(gammahat[#][] (V @@ Take[#, 2])^2 *
            1.51926689 10^14)&,
        Map[process[#]&, process[]], {2}],
    {1}]
```

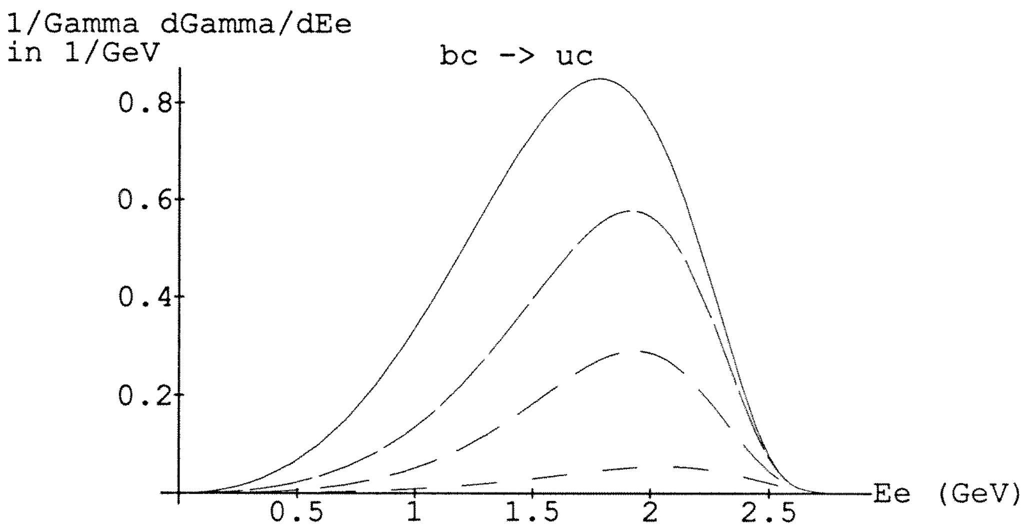
```
{{-b, -c}, {-b, -u}, {c, s}, {c, d}}
```

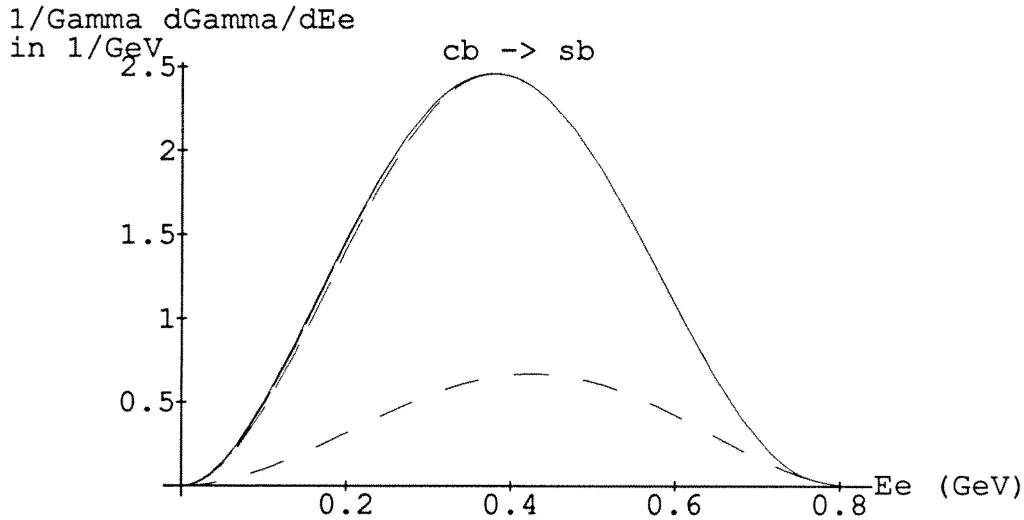
```
{10.5826, 0.113418, 8.87712, 0.543171}
```

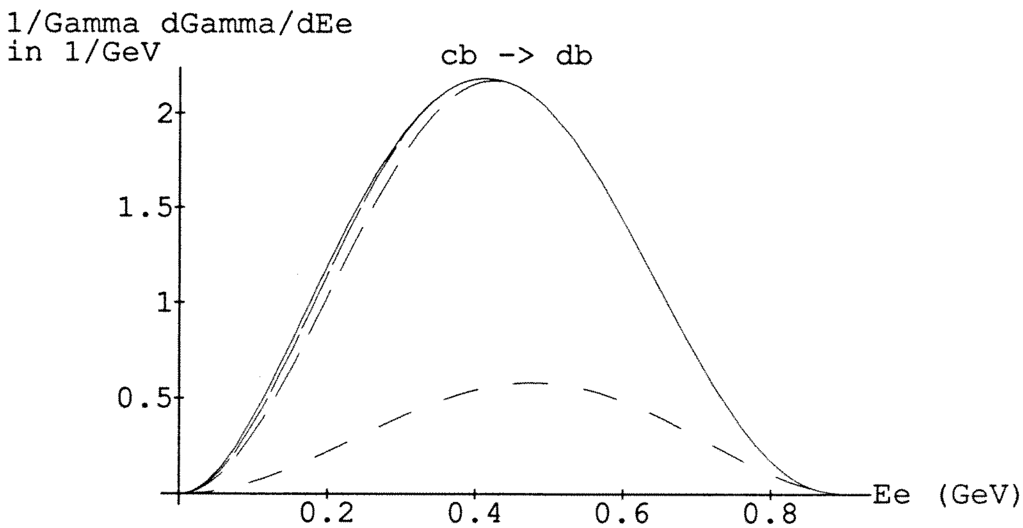
Plot the processes broken into 11S0, 13S1, 1P, and 2S channels:

```
Block[ {i, proc, gtot},
    Do[
    proc = process[process[][[i]]] ;
    gtot = mB/GeV Plus @@ (gammahat[#][]& /@ proc) ;
    gammaplot[process[][[i]]] = Plot[
    Release[{gammahat[proc[[1]]][Ee/mB GeV]/gtot,
        Apply[Plus, Map[Apply[#,{Ee/mB GeV}]&,
            Map[gammahat, proc[[Range[2]]]]]]/gtot,
        Apply[Plus, Map[Apply[#,{Ee/mB GeV}]&,
            Map[gammahat, proc[[Range[6]]]]]]/gtot,
        Apply[Plus, Map[Apply[#,{Ee/mB GeV}]&,
            Map[gammahat, proc]]]/gtot}],
    Release[{Ee, 0,
        mB/GeV Max @@ (xupper /@ proc)}],
    AxesLabel -> {"Ee (GeV)",
        "1/Gamma dGamma/dEe\nin 1/GeV"},
    PlotLabel -> (process[][[i]] /. -p_ :> p /.
        {p_, q_, r_} :> StringJoin @@
        ToString /@ {p, r, " -> ", q, r}),
    PlotStyle ->
        {{Thickness[.001], Dashing[{.05, .05}]},
        {Thickness[.001], Dashing[{.07, .03}]},
        {Thickness[.001], Dashing[{.09, .01}]},
        {Thickness[.001]}},
    PlotRange -> All]
    , {i, Length[process[]]}]
]
```


1/Gamma dGamma/dEe
in 1/GeV
bc -> cc
0.8
0.6
0.4
0.2
0.5
1
1.5
2
Ee (GeV)


1/Gamma dGamma/dEe
in 1/GeV
bc -> uc
0.8
0.6
0.4
0.2
0.5
1
1.5
2
2.5
Ee (GeV)


1/Gamma dGamma/dEe
in 1/GeV
cb -> sb
2.5
2
1.5
1
0.5
0.2
0.4
0.6
0.8
Ee (GeV)


1/Gamma dGamma/dEe
in 1/GeV
cb -> db
2
1.5
1
0.5
0.2
0.4
0.6
0.8
Ee (GeV)

## ■ Cuts on charmonia decays

Calculate total decay rate for electron energies above the highest possible for the lowest mass final state not included in the calculation (the 13D1 state in these cases). The returned set is the lower bound on the electron energy, the decay rate (in 10^10 s^-1), and the fraction of the total rate.

Compute what x (Ee) cut has fraction f of the phase space above the cut with z==mX/mB.

```
phcut[f_,z_] := If[f==1,0,Block[{x,t,z2},
    z2 = z^2 ;
    t = 4/(1 - z2^2 + 2 z2 Log[z2]) ;
    FindRoot[1-t(x(x+z2)+z2 Log[1-2x]/2)==f,
        {x, (1-z)/2,0,1/2}][[1,2]]]]
```

```
m[{-c,c,h13D1}] := 3.77 GeV
m[{-u,c,h13D1}] := 2.82 GeV
```

```
gammaupper[proc_,f_] :=
    ({mB phcut[f,m[Append[Drop[proc,1],h13D1]]/mB],
        1.51926689 10^14 #,
        #/(Plus @@ (gammahat[#][]& /@ process[proc]))})&[
    Plus @@ (
    (nint[gammahat[#][x][[2]],
    {x, phcut[f,m[Append[Drop[proc,1],h13D1]]/mB],
        xupper[#]}])& /@
    process[proc])]
```

Calculate the upper decay rates for b->c and b->u. For f==0, the cut excludes the 13D1 state completely, while for f==0.1, the top 10% of the phase space to the 13D1 state is included. Typically, the top 10% of the phase space accounts for 3% to 4% of the decays.

```
gammaupper[{-b,-c,c},0]
```

```
{2.0016 GeV, 164.705, 0.0358591}
```

```
gammaupper[{-b,-c,c},0.1]
```

```
{1.68894 GeV, 883.257, 0.1923}
```

```
gammaupper[{-b,-u,c},0]
```

```
{2.50084 GeV, 20.8675, 0.00516822}
```

```
gammaupper[{-b,-u,c},0.1]
```

```
{2.16226 GeV, 447.931, 0.110938}
```

## ■ Polarization

Compute total polarization with no Ee cut:

```
gammahatLt[pr_] := nint[gammahatLX[pr][0,y],{y,0,#}]&[
    0.999999 yupper[pr][]]
```

```
gammahatTt[pr_] := nint[gammahatTX[pr][0,y],{y,0,#}]&[
    0.999999 yupper[pr][]]
```

If the integrations are done correctly, the first two should add up to the third.

```
1.51926689 10^14 {gammahatLt[#],gammahatTt[#],gammahat[#][]}&[
    {-b,-c,c,h13S1}]
```

```
{1389.02, 1354.57, 2743.59}
```

# Appendix B

# Results

What follows are the resulting differential decay rate formulas for all the allowed processes, given a $B_c^{\pm}$ mass of 6.27 GeV. Each process is labeled by `{p,q,r,ho}` where the meson $pr$ decays into the meson $qr$ (so the quark decay is $p \to q$), and the harmonic oscillator state of the final meson is $ho$. The harmonic oscillator states are labeled `h11S0` for $1^1\mathrm{S}_0$, `h13S1` for $1^3\mathrm{S}_1$, etc. A minus sign in front of the quark denotes the anti-quark.

The differential rate $d\Gamma/dxdy$ is given as `gammahat[proc][x_,y_]`, the rate integrated over $y$ is given as `gammahat[proc][x_]`, and the rate integrated over $x$ and $y$ is given as `gammahat[proc][]`. For each process, the upper limit for $x = E_e/m_B$ is given as `xupper[proc]` and the upper limit for $y = 1 + m_X^2/m_B^2 - 2E_X/m_B$ as a function of $x$ as `yupper[proc][x_]`, and the maximum of that over $x$ as `yupper[proc][]`.

For the $1^3\mathrm{S}_1$ states, the longitudinal and transverse rates are given as `gammahatL[proc][x_,y_]` and `gammahatT[proc][x_,y_]`. The indefinite integral of those rates over $x$ is given as gammahatLI[{p,q,r,h13S1}][x_,y_] and gammahatTI[proc][x_,y_]. Those are used to compute the integral of the differential longitudinal or transverse rate over $x$ from **xlb** to the upper bound of $x$ by gammahatLX[proc][xlb_,y_] and gammahatTX[proc][xlb_,y_].

```
gammahat/: gammahat[{-b, -c, c, h11S0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3870548852921051 &&
     y < 2*(1 - 0.2258902294157897/(1 - 2*x))*x,
    8.17346879584357*10^-10*E^(4.779215211735137*y)*
     (-4*x^2 - y + 2*x*(0.7741097705842103 + y)), 0]

gammahat/: gammahat[{-b, -c, c, h11S0}][x_] =
   If[x > 0 && x < 0.3870548852921051,
    -(3.578435830982221*10^-11) - 1.932095691077538*10^-10*x +
     6.840845983059317*10^-10*x^2 +
     E^(9.55843042347027*(1 - 0.2258902294157897/(1 - 2*x))*x)*
      (3.578435830982221*10^-11 - 1.48832730045212*10^-10*x +
        (7.726401342556765*10^-11*x)/(1 - 2*x) -
        (1.545280268511353*10^-10*x^2)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -c, c, h11S0}][] = 4.938016881868203*10^-12

gammahat/: gammahat[{-b, -c, c, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3777754884528997 &&
     y < 2*(1 - 0.2444490230942007/(1 - 2*x))*x,
    0.2947490509744869*E^(4.779215211735137*y)*
     (4.690105157017216*10^-9*x - 1.241505947414576*10^-8*x^2 +
       3.787589599020838*10^-9*y - 9.50873465517198*10^-9*x*y -
       4.938342141077321*10^-10*x^2*y + 7.571438005201995*10^-10*y^2 +
       1.889215743569982*10^-9*x*y^2 - 4.347287441124901*10^-9*x^2*y^2 +
       1.567418783445397*10^-10*y^3 + 2.173643720562451*10^-9*x*y^3), 0]

gammahat/: gammahat[{-b, -c, c, h13S1}][x_] =
   If[x > 0 && x < 0.3777754884528997,
    4.531930469341843*10^-11 - 4.147922110668529*10^-10*x +
     7.827790090394253*10^-10*x^2 +
     E^(9.55843042347027*(1 - 0.2444490230942007/(1 - 2*x))*x)*
      (-(4.531930469341843*10^-11) + 8.47973631818943*10^-10*x -
        (1.058907751254063*10^-10*x)/(1 - 2*x) - 1.82022376663156*10^-9*x^2 +
        (9.71081658033712*10^-12*x^2)/(1 - 2*x)^2 +
        (2.138770337676246*10^-10*x^2)/(1 - 2*x) +
        3.702780670230897*10^-10*x^3 -
        (1.129628861269731*10^-12*x^3)/(1 - 2*x)^3 +
        (2.159924971674169*10^-11*x^3)/(1 - 2*x)^2 -
        (1.599687861157617*10^-10*x^3)/(1 - 2*x) -
```

```
        (1.566531361495962*10^-11*x^4)/(1 - 2*x)^3 +
        (1.2816834705797*10^-10*x^4)/(1 - 2*x)^2 -
        (2.621576176407528*10^-10*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -c, c, h13S1}][] = 1.805865466265517*10^-11

gammahat/: gammahat[{-b, -c, c, h11P1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3433071587188938 &&
     y < 2*(1 - 0.3133856825622124/(1 - 2*x))*x,
    0.3402450730667931*E^(5.563800893014054*y)*
     (3.50628266619664*10^-9*x - 1.021325241011839*10^-8*x^2 -
       2.398316659287891*10^-9*y - 8.24474378951836*10^-9*x*y +
       3.801526728238662*10^-8*x^2*y + 9.5405489108961*10^-9*y^2 -
       1.477488861408829*10^-8*x*y^2 - 1.232932352153735*10^-8*x^2*y^2 -
       3.067719003922353*10^-9*y^3 + 6.164661760768673*10^-9*x*y^3), 0]

gammahat/: gammahat[{-b, -c, c, h11P1}][x_] =
   If[x > 0 && x < 0.3433071587188938,
    -(7.05907126904176*10^-11) - 2.335326535176716*10^-10*x +
     1.091124627480364*10^-9*x^2 +
     E^(11.12760178602811*(1 - 0.3133856825622124/(1 - 2*x))*x)*
      (7.05907126904176*10^-11 - 5.519726870932162*10^-10*x +
        (2.461661273236063*10^-10*x)/(1 - 2*x) +
        1.434572265651058*10^-9*x^2 +
        (2.68936378551574*10^-10*x^2)/(1 - 2*x)^2 -
        (1.649681460703355*10^-9*x^2)/(1 - 2*x) -
        7.364545647964568*10^-10*x^3 +
        (4.61916294617647*10^-11*x^3)/(1 - 2*x)^3 -
        (8.769869533492*10^-10*x^3)/(1 - 2*x)^2 +
        (2.558888819742517*10^-9*x^3)/(1 - 2*x) -
        (9.28232903491*10^-11*x^4)/(1 - 2*x)^3 +
        (5.923901155291167*10^-10*x^4)/(1 - 2*x)^2 -
        (9.45145468493949*10^-10*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -c, c, h11P1}][] = 2.088538157597194*10^-12

gammahat/: gammahat[{-b, -c, c, h13P2}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3388111790277492 &&
     y < 2*(1 - 0.3223776419445017/(1 - 2*x))*x,
    0.3536732788291695*E^(5.563800893014054*y)*
     (1.143760190651087*10^-9*x - 3.375804168956927*10^-9*x^2 +
       6.970096609270667*10^-10*y - 7.049162101307135*10^-9*x*y +
       1.474878518030698*10^-8*x^2*y - 4.996004473733421*10^-9*y^2 +
       1.668005950464943*10^-8*x*y^2 - 7.415949601326777*10^-9*x^2*y^2 +
       6.740761675973083*10^-10*y^3 - 6.392589789110507*10^-9*x*y^3 +
       5.771491929609728*10^-9*x^2*y^3 + 2.449655908428487*10^-10*y^4 -
       2.097750870394243*10^-9*x*y^4 - 2.325764742095725*10^-9*x^2*y^4 +
       2.999289730967341*10^-11*y^5 + 1.162882371047862*10^-9*x*y^5), 0]
```

```
gammahat/: gammahat[{-b, -c, c, h13P2}][x_] =
   If[x > 0 && x < 0.3388111790277492,
    2.962726582934688*10^-11 - 2.308989493705753*10^-10*x +
     4.300354576023595*10^-10*x^2 +
     E^(11.12760178602811*(1 - 0.3223776419445017/(1 - 2*x))*x)*
      (-(2.962726582934688*10^-11) + 5.605793655283453*10^-10*x -
        (1.062815951562239*10^-10*x)/(1 - 2*x) -
        3.531600994883173*10^-9*x^2 -
        (1.393924264506761*10^-10*x^2)/(1 - 2*x)^2 +
        (1.432264012717713*10^-9*x^2)/(1 - 2*x) +
        7.46833656372519*10^-9*x^3 -
        (8.81437636531975*10^-12*x^3)/(1 - 2*x)^3 +
        (5.816925288927375*10^-10*x^3)/(1 - 2*x)^2 -
        (4.127194215586275*10^-9*x^3)/(1 - 2*x) -
        4.786260846738087*10^-9*x^4 +
        (2.394918182783101*10^-12*x^4)/(1 - 2*x)^4 +
        (4.070408754318199*10^-11*x^4)/(1 - 2*x)^3 -
        (8.19080111760932*10^-10*x^4)/(1 - 2*x)^2 +
        (3.620589798000928*10^-9*x^4)/(1 - 2*x) +
        6.498792325905174*10^-10*x^5 -
        (2.124338104314747*10^-13*x^5)/(1 - 2*x)^5 -
        (3.122960910306253*10^-11*x^5)/(1 - 2*x)^4 +
        (2.811091607581894*10^-10*x^5)/(1 - 2*x)^3 -
        (7.495878607635434*10^-10*x^5)/(1 - 2*x)^2 +
        (3.626048649526572*10^-10*x^5)/(1 - 2*x) -
        (8.23646780818373*10^-12*x^6)/(1 - 2*x)^5 +
        (1.021965141069139*10^-10*x^6)/(1 - 2*x)^4 -
        (4.755130356923482*10^-10*x^6)/(1 - 2*x)^3 +
        (9.83345769729712*10^-10*x^6)/(1 - 2*x)^2 -
        (7.625728662496681*10^-10*x^6)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -c, c, h13P2}][] = 1.188521698075957*10^-12

gammahat/: gammahat[{-b, -c, c, h13P1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3433071587188938 &&
     y < 2*(1 - 0.3133856825622124/(1 - 2*x))*x,
    0.3402450730667931*E^(5.563800893014054*y)*
     (1.584623656071581*10^-10*x - 4.615760597550194*10^-10*x^2 -
       3.44250533140191*10^-11*y - 5.797868499459851*10^-10*x*y +
       1.986380649204101*10^-8*x^2*y + 3.226536921873298*10^-10*y^2 -
       2.503863260983023*10^-8*x*y^2 - 1.432721068853047*10^-8*x^2*y^2 +
       6.910041461499678*10^-9*y^3 + 7.163605344265233*10^-9*x*y^3), 0]

gammahat/: gammahat[{-b, -c, c, h13P1}][x_] =
   If[x > 0 && x < 0.3433071587188938,
    1.306782324150645*10^-11 + 9.81257342583065*10^-11*x +
     3.031626089360583*10^-10*x^2 +
```

```
      E^(11.12760178602811*(1 - 0.3133856825622124/(1 - 2*x))*x)*
       (-(1.306782324150645*10^-11) + 4.72877989833803*10^-11*x -
         (4.557051936872897*10^-11*x)/(1 - 2*x) +
         6.409570951394237*10^-11*x^2 -
         (8.17581380661771*10^-11*x^2)/(1 - 2*x)^2 +
         (1.457931422119442*10^-10*x^2)/(1 - 2*x) -
         6.296800381672091*10^-10*x^3 -
         (1.040467116926016*10^-10*x^3)/(1 - 2*x)^3 +
         (3.017127679688593*10^-10*x^3)/(1 - 2*x)^2 +
         (2.94005077068757*10^-10*x^3)/(1 - 2*x) -
         (1.078647044431167*10^-10*x^4)/(1 - 2*x)^3 +
         (6.883831039199037*10^-10*x^4)/(1 - 2*x)^2 -
         (1.098300181252294*10^-9*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -c, c, h13P1}][] = 7.086952249335827*10^-13

gammahat/: gammahat[{-b, -c, c, h13P0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3512396694214875 &&
     y < 2*(1 - 0.2975206611570249/(1 - 2*x))*x,
    1.897372681034369*10^-10*E^(5.563800893014054*y)*
     (-4*x^2 - y + 2*x*(0.7024793388429752 + y)), 0]

gammahat/: gammahat[{-b, -c, c, h13P0}][x_] =
   If[x > 0 && x < 0.3512396694214875,
    -(6.129280310947738*10^-12) - 3.565347405157032*10^-11*x +
     1.364083810703379*10^-10*x^2 +
     E^(11.12760178602811*(1 - 0.2975206611570249/(1 - 2*x))*x)*
      (6.129280310947738*10^-12 - 3.255071648359864*10^-11*x +
        (2.029215586170317*10^-11*x)/(1 - 2*x) -
        (4.058431172340634*10^-11*x^2)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -c, c, h13P0}][] = 6.208412854103057*10^-13

gammahat/: gammahat[{-b, -c, c, h21S0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3360830668813544 &&
     y < 2*(1 - 0.3278338662372912/(1 - 2*x))*x,
    2.953165712693572*10^-9*E^(4.779215211735137*y)*
     (0.276504527962585 + 0.02552325902144797*
         (7.182399999999999 - 39.31289999999999*y))^2*
     (-4*x^2 - y + 2*x*(0.672166133762709 + y)), 0]

gammahat/: gammahat[{-b, -c, c, h21S0}][x_] =
   If[x > 0 && x < 0.3360830668813544,
    -(1.114592024372523*10^-10) - 1.863394928637124*10^-10*x +
     1.217728407253243*10^-9*x^2 +
     E^(9.55843042347027*(1 - 0.3278338662372912/(1 - 2*x))*x)*
      (1.114592024372523*10^-10 - 8.79035538688252*10^-10*x +
        (3.492660155863564*10^-10*x)/(1 - 2*x) +
```

```
        2.522823044862461*10^-9*x^2 +
        (4.13009355064754*10^-10*x^2)/(1 - 2*x)^2 -
        (2.486092409446072*10^-9*x^2)/(1 - 2*x) -
        2.672988869315277*10^-9*x^3 +
        (1.753577976737557*10^-10*x^3)/(1 - 2*x)^3 -
        (2.071173229456353*10^-9*x^3)/(1 - 2*x)^2 +
        (5.562434673102687*10^-9*x^3)/(1 - 2*x) -
        (3.507155953475115*10^-10*x^4)/(1 - 2*x)^3 +
        (2.139593443306179*10^-9*x^4)/(1 - 2*x)^2 -
        (3.263228213520669*10^-9*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -c, c, h21S0}][] = 1.116721485345488*10^-12

gammahat/: gammahat[{-b, -c, c, h23S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3268240195966209 &&
     y < 2*(1 - 0.3463519608067582/(1 - 2*x))*x,
    0.4452099110480616*E^(4.779215211735137*y)*
     (1.310302739763531*10^-9*x - 4.009199634043906*10^-9*x^2 +
       1.042228138364186*10^-9*y - 1.102204358101589*10^-8*x*y +
       2.251741669949024*10^-8*x^2*y - 5.480651982894528*10^-9*y^2 +
       2.856892208883502*10^-8*x*y^2 - 3.23061338021647*10^-8*x^2*y^2 +
       6.879980513298923*10^-9*y^3 - 2.409880467302799*10^-8*x*y^3 +
       9.47265316927212*10^-9*x^2*y^3 + 1.248223434256394*10^-9*y^4 -
       2.393406422611792*10^-9*x*y^4 - 7.168751442797849*10^-9*x^2*y^4 +
       1.113323384454093*10^-9*y^5 + 3.584375721398924*10^-9*x*y^5), 0]

gammahat/: gammahat[{-b, -c, c, h23S1}][x_] =
   If[x > 0 && x < 0.3268240195966209,
    9.98894183227859*10^-11 - 6.669995451856731*10^-10*x +
     1.155124587317254*10^-9*x^2 +
     E^(9.55843042347027*(1 - 0.3463519608067582/(1 - 2*x))*x)*
      (-(9.98894183227859*10^-11) + 1.621785600264939*10^-9*x -
        (3.306920223276532*10^-10*x)/(1 - 2*x) - 9.99898124487012*10^-9*x^2 -
        (4.360663285921808*10^-10*x^2)/(1 - 2*x)^2 +
        (4.32211368570147*10^-9*x^2)/(1 - 2*x) +
        2.762599354005725*10^-8*x^3 -
        (2.108657407958021*10^-10*x^3)/(1 - 2*x)^3 +
        (3.635388240990515*10^-9*x^3)/(1 - 2*x)^2 -
        (1.830673729246065*10^-8*x^3)/(1 - 2*x) -
        2.966025372842847*10^-8*x^4 +
        (1.79027106732925*10^-12*x^4)/(1 - 2*x)^4 +
        (5.663028918032016*10^-10*x^4)/(1 - 2*x)^3 -
        (6.8729070711383*10^-9*x^4)/(1 - 2*x)^2 +
        (2.53527288761959*10^-8*x^4)/(1 - 2*x) +
        5.693026332399738*10^-9*x^5 -
        (1.654122994738168*10^-11*x^5)/(1 - 2*x)^5 +
        (1.0702558158011*10^-10*x^5)/(1 - 2*x)^4 -
        (3.362203959417097*10^-10*x^5)/(1 - 2*x)^3 +
```

```
        (1.540373458091003*10^-9*x^5)/(1 - 2*x)^2 -
        (5.042899515036025*10^-9*x^5)/(1 - 2*x) -
        (5.325495166397131*10^-11*x^6)/(1 - 2*x)^5 +
        (6.150385467999022*10^-10*x^6)/(1 - 2*x)^4 -
        (2.663642550343697*10^-9*x^6)/(1 - 2*x)^3 +
        (5.12704387783123*10^-9*x^6)/(1 - 2*x)^2 -
        (3.700746969851705*10^-9*x^6)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -c, c, h23S1}][] = 1.51253804985761*10^-12

gammahat/: gammahat[{-b, -u, c, h11S0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4559991758430439 &&
     y < 2*(1 - 0.0880016483139123/(1 - 2*x))*x,
    1.636692483772921*10^-11*E^(10.79844153223308*y)*
     (-4*x^2 - y + 2*x*(0.911998351686088 + y)), 0]

gammahat/: gammahat[{-b, -u, c, h11S0}][x_] =
   If[x > 0 && x < 0.4559991758430439,
    -(1.403605061645236*10^-13) - 2.483864678971603*10^-12*x +
     6.062698877008997*10^-12*x^2 +
     E^(21.59688306446617*(1 - 0.0880016483139123/(1 - 2*x))*x)*
      (1.403605061645236*10^-13 - 5.474847595328954*10^-13*x +
        (2.667637472038483*10^-13*x)/(1 - 2*x) -
        (5.335274944076966*10^-13*x^2)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -u, c, h11S0}][] = 1.537659186900548*10^-12

gammahat/: gammahat[{-b, -u, c, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4486161031111925 &&
     y < 2*(1 - 0.102767793777615/(1 - 2*x))*x,
    0.006841345611553671*E^(10.79844153223308*y)*
     (7.894477721831866*10^-10*x - 1.759740158028872*10^-9*x^2 +
       4.995356596174423*10^-9*y - 1.713766978021184*10^-8*x*y +
       1.382882147769821*10^-8*x^2*y + 1.419597579127249*10^-9*y^2 -
       1.885563126299301*10^-9*x*y^2 - 1.120969037373893*10^-8*x^2*y^2 -
       5.394644336210646*10^-10*y^3 + 5.604845186869465*10^-9*x*y^3), 0]

gammahat/: gammahat[{-b, -u, c, h13S1}][x_] =
   If[x > 0 && x < 0.4486161031111925,
    2.760251896062271*10^-13 - 1.468218940791095*10^-12*x +
     2.048035024346284*10^-12*x^2 +
     E^(21.59688306446617*(1 - 0.102767793777615/(1 - 2*x))*x)*
      (-(2.760251896062271*10^-13) + 7.429502683563885*10^-12*x -
        (6.126279783271229*10^-13*x)/(1 - 2*x) -
        1.897786979670985*10^-11*x^2 +
        (4.200564396745604*10^-14*x^2)/(1 - 2*x)^2 +
        (1.331098499664139*10^-12*x^2)/(1 - 2*x) +
        8.69451880821672*10^-12*x^3 +
```

```
        (2.967593105375696*10^-15*x^3)/(1 - 2*x)^3 -
        (1.787709270446941*10^-13*x^3)/(1 - 2*x)^2 +
        (5.650556451430485*10^-13*x^3)/(1 - 2*x) -
        (3.083224564334375*10^-14*x^4)/(1 - 2*x)^3 +
        (6.000371227208281*10^-13*x^4)/(1 - 2*x)^2 -
        (2.919383109553185*10^-12*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -u, c, h13S1}][] = 6.118067980332768*10^-12

gammahat/: gammahat[{-b, -u, c, h11P1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4242793078099046 &&
     y < 2*(1 - 0.1514413843801907/(1 - 2*x))*x,
    0.01470129519442256*E^(11.30918808649835*y)*
     (1.214010875444067*10^-8*x - 2.861348298390258*10^-8*x^2 -
       6.21916585119681*10^-9*y - 1.011146725147644*10^-8*x*y +
       5.250972136735685*10^-8*x^2*y + 1.314572861784311*10^-8*y^2 -
       1.984911795902092*10^-8*x*y^2 - 1.509793809583464*10^-8*x^2*y^2 -
       3.55017573234231*10^-9*y^3 + 7.548969047917318*10^-9*x*y^3), 0]

gammahat/: gammahat[{-b, -u, c, h11P1}][x_] =
   If[x > 0 && x < 0.4242793078099046,
    -(1.001234638254395*10^-12) - 1.649951733770488*10^-11*x +
     4.353856044230186*10^-11*x^2 +
     E^(22.6183761729967*(1 - 0.1514413843801907/(1 - 2*x))*x)*
      (1.001234638254395*10^-12 - 6.1467843477673*10^-12*x +
        (3.429587278339353*10^-12*x)/(1 - 2*x) +
        1.34714938650164*10^-11*x^2 +
        (1.679990574093885*10^-12*x^2)/(1 - 2*x)^2 -
        (1.972702027408865*10^-11*x^2)/(1 - 2*x) -
        7.082692167174904*10^-12*x^3 +
        (1.282323856801964*10^-13*x^3)/(1 - 2*x)^3 -
        (5.146135649958944*10^-12*x^3)/(1 - 2*x)^2 +
        (2.946240637326921*10^-11*x^3)/(1 - 2*x) -
        (2.726688432974341*10^-13*x^4)/(1 - 2*x)^3 +
        (3.600981916711805*10^-12*x^4)/(1 - 2*x)^2 -
        (1.188902865438554*10^-11*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -u, c, h11P1}][] = 3.779365375662306*10^-12

gammahat/: gammahat[{-b, -u, c, h13P2}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4205095528439774 &&
     y < 2*(1 - 0.1589808943120452/(1 - 2*x))*x,
    0.01676198715898141*E^(11.30918808649835*y)*
     (8.50242643840626*10^-10*x - 2.021934194099256*10^-9*x^2 +
       1.527490761997584*10^-9*y - 7.538924331049623*10^-9*x*y +
       8.91295841550944*10^-9*x^2*y - 5.184169416031771*10^-9*y^2 +
       1.879863879638239*10^-8*x*y^2 - 1.787976452616761*10^-8*x^2*y^2 +
       1.703870470971734*10^-9*y^3 - 3.251104752392807*10^-9*x*y^3 +
```

```
      1.28464283593743*10^-8*x^2*y^3 - 6.065401353213056*10^-10*y^4 -
      5.09323452070205*10^-9*x*y^4 - 3.162781083069868*10^-9*x^2*y^4 +
      3.990099575157005*10^-10*y^5 + 1.581390541534934*10^-9*x*y^5), 0]

gammahat/: gammahat[{-b, -u, c, h13P2}][x_] =
   If[x > 0 && x < 0.4205095528439774,
    3.325224898327152*10^-13 - 2.691319521914669*10^-12*x +
     4.665198993357307*10^-12*x^2 +
     E^(22.6183761729967*(1 - 0.1589808943120452/(1 - 2*x))*x)*
      (-(3.325224898327152*10^-13) + 1.021243828293249*10^-11*x -
        (1.195714186853715*10^-12*x)/(1 - 2*x) -
        7.088514939871244*10^-11*x^2 -
        (8.55560894356698*10^-13*x^2)/(1 - 2*x)^2 +
        (1.590923971814351*10^-11*x^2)/(1 - 2*x) +
        1.745638025404262*10^-10*x^3 -
        (9.43753439390926*10^-14*x^3)/(1 - 2*x)^3 +
        (4.645567438783203*10^-12*x^3)/(1 - 2*x)^2 -
        (5.323927441964651*10^-11*x^3)/(1 - 2*x) -
        1.607892413250921*10^-10*x^4 -
        (1.186117477348824*10^-14*x^4)/(1 - 2*x)^4 +
        (3.557170678569357*10^-13*x^4)/(1 - 2*x)^3 -
        (7.131043457812247*10^-12*x^4)/(1 - 2*x)^2 +
        (5.929506565284433*10^-11*x^4)/(1 - 2*x) +
        4.714840066005855*10^-11*x^5 -
        (1.921995945685513*10^-15*x^5)/(1 - 2*x)^5 -
        (2.730361234218241*10^-14*x^5)/(1 - 2*x)^4 +
        (7.820328059782489*10^-13*x^5)/(1 - 2*x)^3 -
        (3.492329902638697*10^-12*x^5)/(1 - 2*x)^2 -
        (6.666177834252297*10^-12*x^5)/(1 - 2*x) -
        (7.617419445618633*10^-15*x^6)/(1 - 2*x)^5 +
        (1.916562233111429*10^-13*x^6)/(1 - 2*x)^4 -
        (1.808294865938071*10^-12*x^6)/(1 - 2*x)^3 +
        (7.582860291747925*10^-12*x^6)/(1 - 2*x)^2 -
        (1.192416913453828*10^-11*x^6)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -u, c, h13P2}][] = 6.330322599174957*10^-13

gammahat/: gammahat[{-b, -u, c, h13P1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4211442045740711 &&
     y < 2*(1 - 0.1577115908518578/(1 - 2*x))*x,
    0.01640185568542311*E^(11.30918808649835*y)*
     (3.786121025174936*10^-9*x - 8.99008221899687*10^-9*x^2 -
       1.838344514868387*10^-9*y - 9.46417928197406*10^-10*x*y +
       3.700852145751728*10^-8*x^2*y + 2.357607469493635*10^-9*y^2 -
       2.394746150632641*10^-8*x*y^2 - 2.016523860080056*10^-8*x^2*y^2 +
       2.968610422420562*10^-9*y^3 + 1.008261930040028*10^-8*x*y^3), 0]

gammahat/: gammahat[{-b, -u, c, h13P1}][x_] =
```

```
   If[x > 0 && x < 0.4211442045740711,
    -(2.713620661852714*10^-13) - 5.008659732091322*10^-12*x +
     1.824180536427656*10^-11*x^2 +
     E^(22.6183761729967*(1 - 0.1577115908518578/(1 - 2*x))*x)*
      (2.713620661852714*10^-13 - 1.129109559968774*10^-12*x +
        (9.67997359332479*10^-13*x)/(1 - 2*x) +
        1.777930507192639*10^-12*x^2 +
        (2.265593189691176*10^-13*x^2)/(1 - 2*x)^2 -
        (4.593886359901526*10^-12*x^2)/(1 - 2*x) -
        2.306125557306174*10^-12*x^3 -
        (1.351125192740755*10^-13*x^3)/(1 - 2*x)^3 -
        (1.271293053314584*10^-12*x^3)/(1 - 2*x)^2 +
        (1.385668304110837*10^-11*x^3)/(1 - 2*x) -
        (4.588975650930002*10^-13*x^4)/(1 - 2*x)^3 +
        (5.819452617456044*10^-12*x^4)/(1 - 2*x)^2 -
        (1.844966684446925*10^-11*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -u, c, h13P1}][] = 3.648198654303766*10^-12

gammahat/: gammahat[{-b, -u, c, h13P0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4267416039010096 &&
     y < 2*(1 - 0.1465167921979808/(1 - 2*x))*x,
    8.56916392554613*10^-12*E^(11.30918808649835*y)*
     (-4*x^2 - y + 2*x*(0.853483207802019 + y)), 0]

gammahat/: gammahat[{-b, -u, c, h13P0}][x_] =
   If[x > 0 && x < 0.4267416039010096,
    -(6.700012251012702*10^-14) - 1.159397204808695*10^-12*x +
     3.030867949141834*10^-12*x^2 +
     E^(22.6183761729967*(1 - 0.1465167921979808/(1 - 2*x))*x)*
      (6.700012251012702*10^-14 - 3.560367697622212*10^-13*x +
        (2.220365247419672*10^-13*x)/(1 - 2*x) -
        (4.440730494839343*10^-13*x^2)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -u, c, h13P0}][] = 3.219278005342186*10^-13

gammahat/: gammahat[{-b, -u, c, h21S0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4153407660081042 &&
     y < 2*(1 - 0.1693184679837916/(1 - 2*x))*x,
    9.55455502565201*10^-10*E^(10.79844153223308*y)*
     (0.3840085806253211 + 0.01230619273988189*
         (13.6161 - 39.31289999999999*y))^2*
     (-4*x^2 - y + 2*x*(0.830681532016208 + y)), 0]

gammahat/: gammahat[{-b, -u, c, h21S0}][x_] =
   If[x > 0 && x < 0.4153407660081042,
    -(3.401431837859712*10^-12) - 4.577391110660785*10^-11*x +
     1.265870800202198*10^-10*x^2 +
```

```
    E^(21.59688306446617*(1 - 0.1693184679837916/(1 - 2*x))*x)*
     (3.401431837859712*10^-12 - 2.768641454740061*10^-11*x +
       (1.243818979732714*10^-11*x)/(1 - 2*x) +
       6.258445487505242*10^-11*x^2 +
       (6.07489630924211*10^-12*x^2)/(1 - 2*x)^2 -
       (6.790875606520001*10^-11*x^2)/(1 - 2*x) -
       4.339432705207861*10^-11*x^3 +
       (8.04210893070655*10^-13*x^3)/(1 - 2*x)^3 -
       (2.245339129419936*10^-11*x^3)/(1 - 2*x)^2 +
       (1.11906034022573*10^-10*x^3)/(1 - 2*x) -
       (1.608421786141311*10^-12*x^4)/(1 - 2*x)^3 +
       (1.899877556528896*10^-11*x^4)/(1 - 2*x)^2 -
       (5.610367194884988*10^-11*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -u, c, h21S0}][] = 5.225407612458527*10^-12

gammahat/: gammahat[{-b, -u, c, h23S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4113573407202216 &&
     y < 2*(1 - 0.1772853185595568/(1 - 2*x))*x,
    0.02679815516328711*E^(10.79844153223308*y)*
     (1.862396212576178*10^-9*x - 4.52744129791246*10^-9*x^2 +
       1.192957410698045*10^-8*y - 5.396274367645793*10^-8*x*y +
       6.567703718852907*10^-8*x^2*y - 2.159248006774689*10^-8*y^2 +
       7.552619863923021*10^-8*x*y^2 - 1.112675829134272*10^-7*x^2*y^2 +
       1.115922293463219*10^-8*y^3 - 5.094739152234485*10^-9*x*y^3 +
       6.855837479199443*10^-8*x^2*y^3 - 3.466832845250133*10^-9*y^4 -
       2.691967961204504*10^-8*x*y^4 - 1.789078996637533*10^-8*x^2*y^4 +
       1.757878014375804*10^-9*y^5 + 8.94539498318767*10^-9*x*y^5), 0]

gammahat/: gammahat[{-b, -u, c, h23S1}][x_] =
   If[x > 0 && x < 0.4113573407202216,
    3.811416905701988*10^-12 - 2.016237360574742*10^-11*x +
     3.195451451377255*10^-11*x^2 +
     E^(21.59688306446617*(1 - 0.1772853185595568/(1 - 2*x))*x)*
      (-(3.811416905701988*10^-12) + 1.024770988281227*10^-10*x -
        (1.459319228319119*10^-11*x)/(1 - 2*x) -
        6.170709658729725*10^-10*x^2 -
        (7.841478866913565*10^-12*x^2)/(1 - 2*x)^2 +
        (1.479633982073375*10^-10*x^2)/(1 - 2*x) +
        1.432536325239862*10^-9*x^3 -
        (1.40990375150095*10^-12*x^3)/(1 - 2*x)^3 +
        (4.686639019940206*10^-11*x^3)/(1 - 2*x)^2 -
        (4.734649762408518*10^-10*x^3)/(1 - 2*x) -
        1.354555573814361*10^-9*x^4 -
        (1.679105837621569*10^-13*x^4)/(1 - 2*x)^4 +
        (3.079240895803247*10^-12*x^4)/(1 - 2*x)^3 -
        (6.128442504153108*10^-11*x^4)/(1 - 2*x)^2 +
        (5.179883515302316*10^-10*x^4)/(1 - 2*x) +
```

```
        3.989279362373687*10^-10*x^5 -
        (2.444817183905711*10^-14*x^5)/(1 - 2*x)^5 -
        (5.28856293242376*10^-13*x^5)/(1 - 2*x)^4 +
        (1.139353034795344*10^-11*x^5)/(1 - 2*x)^3 -
        (4.79651760089202*10^-11*x^5)/(1 - 2*x)^2 -
        (4.301398693355806*10^-11*x^5)/(1 - 2*x) -
        (1.244105403951298*10^-13*x^6)/(1 - 2*x)^5 +
        (2.807012817665115*10^-12*x^6)/(1 - 2*x)^4 -
        (2.374996001196343*10^-11*x^6)/(1 - 2*x)^3 +
        (8.93097454616542*10^-11*x^6)/(1 - 2*x)^2 -
        (1.259406957486608*10^-10*x^6)/(1 - 2*x)), 0]

gammahat/: gammahat[{-b, -u, c, h23S1}][] = 5.312667724929053*10^-12

gammahat/: gammahat[{c, s, -b, h11S0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.1305016928285626 &&
     y < 2*(1 - 0.7389966143428748/(1 - 2*x))*x,
    1.267347149516607*10^-9*E^(28.58526263869093*y)*
     (-4*x^2 - y + 2*x*(0.2610033856571252 + y)), 0]

gammahat/: gammahat[{c, s, -b, h11S0}][x_] =
   If[x > 0 && x < 0.1305016928285626,
    -(1.550997931858503*10^-12) - 2.004153099540088*10^-11*x +
     1.773427329369671*10^-10*x^2 +
     E^(57.17052527738186*(1 - 0.7389966143428748/(1 - 2*x))*x)*
      (1.550997931858503*10^-12 - 6.862983547308265*10^-11*x +
        (6.552783960936565*10^-11*x)/(1 - 2*x) -
        (1.310556792187313*10^-10*x^2)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, s, -b, h11S0}][] = 1.746085529971109*10^-14

gammahat/: gammahat[{c, s, -b, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.1222295989357183 &&
     y < 2*(1 - 0.7555408021285634/(1 - 2*x))*x,
    0.6132904546580836*E^(28.58526263869093*y)*
     (9.9589897304939*10^-10*x - 8.14777256671801*10^-9*x^2 +
       5.024271169469572*10^-9*y - 4.740940166358013*10^-8*x*y +
       6.044097024555623*10^-8*x^2*y - 1.211536928618664*10^-9*y^2 -
       2.995689584979429*10^-8*x*y^2 - 2.156509350263397*10^-9*x^2*y^2 -
       7.249265434475198*10^-9*y^3 + 1.078254675131698*10^-9*x*y^3), 0]

gammahat/: gammahat[{c, s, -b, h13S1}][x_] =
   If[x > 0 && x < 0.1222295989357183,
    3.874560084975946*10^-12 - 5.537101875370957*10^-11*x +
     2.202860947296078*10^-10*x^2 +
     E^(57.17052527738186*(1 - 0.7555408021285634/(1 - 2*x))*x)*
      (-(3.874560084975946*10^-12) + 2.768816540305617*10^-10*x -
        (1.673603230570805*10^-10*x)/(1 - 2*x) -
```

```
        2.333590241026091*10^-9*x^2 -
        (9.66233190421002*10^-11*x^2)/(1 - 2*x)^2 +
        (1.724573811632972*10^-9*x^2)/(1 - 2*x) +
        1.26363486995018*10^-9*x^3 -
        (5.366384370498133*10^-10*x^3)/(1 - 2*x)^3 +
        (6.577054650002948*10^-10*x^3)/(1 - 2*x)^2 -
        (8.85154810634933*10^-10*x^3)/(1 - 2*x) -
        (7.981952224463115*10^-11*x^4)/(1 - 2*x)^3 +
        (2.11291096443125*10^-10*x^4)/(1 - 2*x)^2 -
        (1.398277206524523*10^-10*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, s, -b, h13S1}][] = 4.330190019030609*10^-14

gammahat/: gammahat[{c, s, -b, h11P1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.06026647741581009 &&
     y < 2*(1 - 0.87946704516838/(1 - 2*x))*x,
    0.884316522789575*E^(31.77604662650889*y)*
     (2.070713047904492*10^-10*x - 3.435928457569465*10^-9*x^2 +
       1.252557227848051*10^-10*y - 7.592055972384071*10^-9*x*y +
       7.378045327298902*10^-8*x^2*y + 1.798963559794618*10^-8*y^2 -
       3.593945312112409*10^-8*x*y^2 - 1.577615875589565*10^-8*x^2*y^2 -
       3.499402798515071*10^-9*y^3 + 7.888079377947826*10^-9*x*y^3), 0]

gammahat/: gammahat[{c, s, -b, h11P1}][x_] =
   If[x > 0 && x < 0.06026647741581009,
    -(9.00166866567257*10^-13) - 1.038972699964831*10^-11*x +
     1.611077328863962*10^-10*x^2 +
     E^(63.55209325301779*(1 - 0.87946704516838/(1 - 2*x))*x)*
      (9.00166866567257*10^-13 - 4.681776164771085*10^-11*x +
        (5.031210100219659*10^-11*x)/(1 - 2*x) +
        1.584195757511209*10^-9*x^2 +
        (1.577367452487066*10^-9*x^2)/(1 - 2*x)^2 -
        (3.328486145919197*10^-9*x^2)/(1 - 2*x) -
        7.008923822533161*10^-10*x^3 +
        (5.299690512684498*10^-10*x^3)/(1 - 2*x)^3 -
        (4.966342837774621*10^-9*x^3)/(1 - 2*x)^2 +
        (5.57821259421028*10^-9*x^3)/(1 - 2*x) -
        (1.194614677120094*10^-9*x^4)/(1 - 2*x)^3 +
        (2.716678660520763*10^-9*x^4)/(1 - 2*x)^2 -
        (1.544502818750097*10^-9*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, s, -b, h11P1}][] = 3.637068070936316*10^-16

gammahat/: gammahat[{c, s, -b, h13P2}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.06026647741581009 &&
     y < 2*(1 - 0.87946704516838/(1 - 2*x))*x,
    0.884316522789575*E^(31.77604662650889*y)*
     (9.88630270689401*10^-12*x - 1.640431485431482*10^-10*x^2 +
```

```
        4.015851834364749*10^-11*y - 2.474682855207748*10^-9*x*y +
        3.205249354041837*10^-8*x^2*y - 1.320873687991936*10^-8*y^2 +
        1.585844944800347*10^-7*x*y^2 - 2.140266692287995*10^-7*x^2*y^2 +
        9.26257307771233*10^-9*y^3 + 6.009220673032159*10^-8*x*y^3 +
        6.471910976525729*10^-8*x^2*y^3 - 2.494676067426515*10^-8*y^4 -
        3.218646477612492*10^-8*x*y^4 - 2.872079370252218*10^-9*x^2*y^4 +
        6.183101678590189*10^-9*y^5 + 1.436039685126109*10^-9*x*y^5), 0]

gammahat/: gammahat[{c, s, -b, h13P2}][x_] =
   If[x > 0 && x < 0.06026647741581009,
    8.28470516459442*10^-13 - 1.08502648821216*10^-11*x +
     4.477367014996214*10^-11*x^2 +
     E^(63.55209325301779*(1 - 0.87946704516838/(1 - 2*x))*x)*
      (-(8.28470516459442*10^-13) + 6.350130040152786*10^-11*x -
        (4.630485063330765*10^-11*x)/(1 - 2*x) -
        2.318861551087046*10^-9*x^2 -
        (1.23909901317552*10^-9*x^2)/(1 - 2*x)^2 +
        (3.408905694647602*10^-9*x^2)/(1 - 2*x) +
        2.232292108837709*10^-8*x^3 -
        (1.896911465503433*10^-9*x^3)/(1 - 2*x)^3 +
        (1.960326241231842*10^-8*x^3)/(1 - 2*x)^2 -
        (3.946971365855216*10^-8*x^3)/(1 - 2*x) -
        2.176335711292946*10^-8*x^4 -
        (6.904556126217059*10^-9*x^4)/(1 - 2*x)^4 +
        (2.168475204914428*10^-8*x^4)/(1 - 2*x)^3 -
        (3.936607849372543*10^-8*x^4)/(1 - 2*x)^2 +
        (4.601574884661592*10^-8*x^4)/(1 - 2*x) +
        5.563304978761334*10^-9*x^5 -
        (2.89709071745585*10^-9*x^5)/(1 - 2*x)^5 +
        (7.836605504385418*10^-9*x^5)/(1 - 2*x)^4 -
        (8.04256212195265*10^-9*x^5)/(1 - 2*x)^3 +
        (9.23291015496255*10^-9*x^5)/(1 - 2*x)^2 -
        (1.167070121531868*10^-8*x^5)/(1 - 2*x) -
        (6.728560288896416*10^-10*x^6)/(1 - 2*x)^5 +
        (3.060289899825951*10^-9*x^6)/(1 - 2*x)^4 -
        (5.219564365666547*10^-9*x^6)/(1 - 2*x)^3 +
        (3.956611673203535*10^-9*x^6)/(1 - 2*x)^2 -
        (1.124718571019911*10^-9*x^6)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, s, -b, h13P2}][] = 1.257337921657375*10^-17

gammahat/: gammahat[{c, s, -b, h13P1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.06026647741581009 &&
     y < 2*(1 - 0.87946704516838/(1 - 2*x))*x,
    0.884316522789575*E^(31.77604662650889*y)*
     (5.147137071893318*10^-11*x - 8.54063036799134*10^-10*x^2 +
       1.424974220110229*10^-11*y + 8.56005331447036*10^-9*x*y +
       7.164088646665005*10^-8*x^2*y - 1.255333118466371*10^-8*y^2 -
```

```
     2.076020275752215*10^-7*x*y^2 - 2.233082424209402*10^-7*x^2*y^2 +
     1.139609304700973*10^-7*y^3 + 1.116541212104701*10^-7*x*y^3), 0]

gammahat/: gammahat[{c, s, -b, h13P1}][x_] =
   If[x > 0 && x < 0.06026647741581009,
    1.297548788840848*10^-12 + 1.808938099992108*10^-11*x +
     9.88213849357242*10^-11*x^2 +
     E^(63.55209325301779*(1 - 0.87946704516838/(1 - 2*x))*x)*
      (-(1.297548788840848*10^-12) + 6.437256062883278*10^-11*x -
        (7.252256014308757*10^-11*x)/(1 - 2*x) -
        1.453281326744496*10^-9*x^2 -
        (2.007221204950334*10^-9*x^2)/(1 - 2*x)^2 +
        (3.47351831007669*10^-9*x^2)/(1 - 2*x) +
        5.858309397021097*10^-9*x^3 -
        (1.725887806586182*10^-8*x^3)/(1 - 2*x)^3 +
        (4.009042803210021*10^-8*x^3)/(1 - 2*x)^2 -
        (2.84233082442573*10^-8*x^3)/(1 - 2*x) -
        (1.690952202279622*10^-8*x^4)/(1 - 2*x)^3 +
        (3.845402079746782*10^-8*x^4)/(1 - 2*x)^2 -
        (2.186211581703187*10^-8*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, s, -b, h13P1}][] = 1.609791853306982*10^-16

gammahat/: gammahat[{c, s, -b, h13P0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.06771314250538618 &&
     y < 2*(1 - 0.864573714989228/(1 - 2*x))*x,
    2.108120677791754*10^-10*E^(31.77604662650889*y)*
     (-4*x^2 - y + 2*x*(0.1354262850107724 + y)), 0]

gammahat/: gammahat[{c, s, -b, h13P0}][x_] =
   If[x > 0 && x < 0.06771314250538618,
    -(2.087832914712135*10^-13) - 1.379352682083808*10^-12*x +
     2.653723041850111*10^-11*x^2 +
     E^(63.55209325301779*(1 - 0.864573714989228/(1 - 2*x))*x)*
      (2.087832914712135*10^-13 - 1.188926252716675*10^-11*x +
        (1.147169594422432*10^-11*x)/(1 - 2*x) -
        (2.294339188844864*10^-11*x^2)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, s, -b, h13P0}][] = 8.98796421186335*10^-17

gammahat/: gammahat[{c, s, -b, h21S0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.04518237016348309 &&
     y < 2*(1 - 0.909635259673034/(1 - 2*x))*x,
    1.042783498433157*10^-8*E^(28.58526263869093*y)*
     (0.4079889018664372 + 0.07551886151341708*
         (0.0840999999999995 - 39.31289999999999*y))^2*
     (-4*x^2 - y + 2*x*(0.0903647403269662 + y)), 0]
```

```
gammahat/: gammahat[{c, s, -b, h21S0}][x_] =
   If[x > 0 && x < 0.04518237016348309,
    -(5.213584124890986*10^-12) - 7.988189956571207*10^-12*x +
     4.075784014809538*10^-10*x^2 +
     E^(57.17052527738186*(1 - 0.909635259673034/(1 - 2*x))*x)*
      (5.213584124890987*10^-12 - 2.900751530412657*10^-10*x +
        (2.711289264068499*10^-10*x)/(1 - 2*x) +
        4.722592126606739*10^-9*x^2 +
        (4.087341874300321*10^-9*x^2)/(1 - 2*x)^2 -
        (9.15996948750822*10^-9*x^2)/(1 - 2*x) -
        2.429849339194024*10^-8*x^3 +
        (1.936086874522438*10^-8*x^3)/(1 - 2*x)^3 -
        (7.010397437356526*10^-8*x^3)/(1 - 2*x)^2 +
        (7.577236953559335*10^-8*x^3)/(1 - 2*x) -
        (3.872173749044877*10^-8*x^4)/(1 - 2*x)^3 +
        (8.51368437594805*10^-8*x^4)/(1 - 2*x)^2 -
        (4.679724255086742*10^-8*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, s, -b, h21S0}][] = 9.49991466037993*10^-17

gammahat/: gammahat[{c, s, -b, h23S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.04060753594875977 &&
     y < 2*(1 - 0.91878492810248/(1 - 2*x))*x,
    0.952035050754517*E^(28.58526263869093*y)*
     (1.147572539420884*10^-10*x - 2.826008800112711*10^-9*x^2 +
       9.24184054040109*10^-10*y - 2.568745944744463*10^-8*x*y +
       1.260073478467141*10^-7*x^2*y - 2.566529841983727*10^-8*y^2 +
       4.067127869357328*10^-7*x*y^2 - 0.000001043003266987813*x^2*y^2 +
       2.170126664592699*10^-7*y^3 - 0.000001522871285812251*x*y^3 +
       0.000002670624957664743*x^2*y^3 - 2.557089202604668*10^-7*y^4 -
       0.000001332150963534427*x*y^4 - 7.785538383647945*10^-8*x^2*y^4 +
       3.224692111445968*10^-7*y^5 + 3.892769191823973*10^-8*x*y^5), 0]

gammahat/: gammahat[{c, s, -b, h23S1}][x_] =
   If[x > 0 && x < 0.04060753594875977,
    5.399237972649965*10^-12 - 7.833113491595268*10^-11*x +
     3.488987946443064*10^-10*x^2 +
     E^(57.17052527738186*(1 - 0.91878492810248/(1 - 2*x))*x)*
      (-(5.399237972649965*10^-12) + 3.870084059099375*10^-10*x -
        (2.836080242370782*10^-10*x)/(1 - 2*x) -
        1.167253470356862*10^-8*x^2 -
        (5.963110534498452*10^-9*x^2)/(1 - 2*x)^2 +
        (1.689419971181271*10^-8*x^2)/(1 - 2*x) +
        1.56878250898925*10^-7*x^3 -
        (5.387191740120408*10^-8*x^3)/(1 - 2*x)^3 +
        (2.374030318170697*10^-7*x^3)/(1 - 2*x)^2 -
        (3.387086533672112*10^-7*x^3)/(1 - 2*x) -
        6.985940224818469*10^-7*x^4 -
```

```
        (1.185220201224343*10^-7*x^4)/(1 - 2*x)^4 +
        (7.919816617687254*10^-7*x^4)/(1 - 2*x)^3 -
        (0.000001892501229510187*x^4)/(1 - 2*x)^2 +
        (0.000001916273976848862*x^4)/(1 - 2*x) +
        3.446348555521608*10^-7*x^5 -
        (2.250185368181098*10^-7*x^5)/(1 - 2*x)^5 +
        (7.160876385704336*10^-7*x^5)/(1 - 2*x)^4 -
        (0.000001006113818634437*x^5)/(1 - 2*x)^3 +
        (0.000001096667400646511*x^5)/(1 - 2*x)^2 -
        (9.25904637464414*10^-7*x^5)/(1 - 2*x) -
        (2.716368563081416*10^-8*x^6)/(1 - 2*x)^5 +
        (1.182591694746841*10^-7*x^6)/(1 - 2*x)^4 -
        (1.930688551654607*10^-7*x^6)/(1 - 2*x)^3 +
        (1.400899886071604*10^-7*x^6)/(1 - 2*x)^2 -
        (3.811827564925373*10^-8*x^6)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, s, -b, h23S1}][] = 5.595834521055685*10^-17

gammahat/: gammahat[{c, d, -b, h11S0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.1454293628808864 &&
     y < 2*(1 - 0.7091412742382273/(1 - 2*x))*x,
    7.6426201988174*10^-10*E^(33.05874942611109*y)*
     (-4*x^2 - y + 2*x*(0.2908587257617726 + y)), 0]

gammahat/: gammahat[{c, d, -b, h11S0}][x_] =
   If[x > 0 && x < 0.1454293628808864,
    -(6.993095133772243*10^-13) - 1.204969835426217*10^-11*x +
     9.24731918961334*10^-11*x^2 +
     E^(66.11749885222218*(1 - 0.7091412742382273/(1 - 2*x))*x)*
      (6.993095133772243*10^-13 - 3.418689759380451*10^-11*x +
        (3.278827856705006*10^-11*x)/(1 - 2*x) -
        (6.557655713410012*10^-11*x^2)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, d, -b, h11S0}][] = 2.002540509828624*10^-14

gammahat/: gammahat[{c, d, -b, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.1386822137262832 &&
     y < 2*(1 - 0.7226355725474336/(1 - 2*x))*x,
    0.4756698807716538*E^(33.05874942611109*y)*
     (7.178061428481286*10^-10*x - 5.175906293685657*10^-9*x^2 +
       4.422279361478407*10^-9*y - 4.127933492336502*10^-8*x*y +
       5.944247526549082*10^-8*x^2*y - 1.762262271546215*10^-9*y^2 -
       2.936202879027068*10^-8*x*y^2 - 2.590157979333197*10^-9*x^2*y^2 +
       6.762269055945049*10^-9*y^3 + 1.295078989666598*10^-9*x*y^3), 0]

gammahat/: gammahat[{c, d, -b, h13S1}][x_] =
   If[x > 0 && x < 0.1386822137262832,
    1.98733233167652*10^-12 - 2.751858327982002*10^-11*x +
```

```
      1.00414348779684*10^-10*x^2 +
      E^(66.11749885222218*(1 - 0.7226355725474336/(1 - 2*x))*x)*
       (-(1.98733233167652*10^-12) + 1.589160264384264*10^-10*x -
         (9.49524665681883*10^-11*x)/(1 - 2*x) - 1.373741677811787*10^-9*x^2 -
         (7.140851253596763*10^-11*x^2)/(1 - 2*x)^2 +
         (1.018968392583225*10^-9*x^2)/(1 - 2*x) +
         7.968172988381877*10^-10*x^3 -
         (2.937376897475552*10^-10*x^3)/(1 - 2*x)^3 +
         (3.334328766323319*10^-10*x^3)/(1 - 2*x)^2 -
         (4.747227099326675*10^-10*x^3)/(1 - 2*x) -
         (5.625530532992088*10^-11*x^4)/(1 - 2*x)^3 +
         (1.556948134496335*10^-10*x^4)/(1 - 2*x)^2 -
         (1.077270614431409*10^-10*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, d, -b, h13S1}][] = 5.114213408332779*10^-14

gammahat/: gammahat[{c, d, -b, h11P1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.07215061722742411 &&
     y < 2*(1 - 0.855698765545152/(1 - 2*x))*x,
    0.821488769971851*E^(34.99488990110552*y)*
     (3.109312605708869*10^-10*x - 4.309474714413167*10^-9*x^2 +
       3.057500280663324*10^-10*y - 9.3001896798741*10^-9*x*y +
       5.640855059001392*10^-8*x^2*y + 1.307470743926398*10^-8*y^2 -
       2.743209864260085*10^-8*x*y^2 - 1.070228754900543*10^-8*x^2*y^2 -
       1.901285649265961*10^-9*y^3 + 5.351143774502716*10^-9*x*y^3), 0]

gammahat/: gammahat[{c, d, -b, h11P1}][x_] =
   If[x > 0 && x < 0.07215061722742411,
    -(3.023963965063549*10^-13) - 1.246827418254993*10^-11*x +
     1.394120044688399*10^-10*x^2 +
     E^(69.98977980221105*(1 - 0.855698765545152/(1 - 2*x))*x)*
      (3.023963965063549*10^-13 - 8.69638302191196*10^-12*x +
        (1.811057104304434*10^-11*x)/(1 - 2*x) +
        7.41784586180172*10^-10*x^2 +
        (9.101464712263*10^-10*x^2)/(1 - 2*x)^2 -
        (1.8176683594545*10^-9*x^2)/(1 - 2*x) -
        2.989065877813833*10^-10*x^3 +
        (2.237160140727819*10^-10*x^3)/(1 - 2*x)^3 -
        (2.70193761170294*10^-9*x^3)/(1 - 2*x)^2 +
        (3.107823366680388*10^-9*x^3)/(1 - 2*x) -
        (6.296458169893164*10^-10*x^4)/(1 - 2*x)^3 +
        (1.471652974953585*10^-9*x^4)/(1 - 2*x)^2 -
        (8.59912994040618*10^-10*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, d, -b, h11P1}][] = 1.134259952610328*10^-15

gammahat/: gammahat[{c, d, -b, h13P2}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.07215061722742411 &&
```

```
    y < 2*(1 - 0.855698765545152/(1 - 2*x))*x,
   0.821488769971851*E^(34.99488990110552*y)*
    (1.014532585540718*10^-11*x - 1.406131540556109*10^-10*x^2 +
      4.189344876004461*10^-11*y - 1.952529603868127*10^-9*x*y +
      1.933893700682712*10^-8*x^2*y - 9.1805849385427*10^-9*y^2 +
      1.146072089871257*10^-7*x*y^2 - 1.722719868887727*10^-7*x^2*y^2 +
      1.114793272293942*10^-8*y^3 + 5.232094948565015*10^-8*x*y^3 +
      5.098940398004283*10^-8*x^2*y^3 - 2.194484296522687*10^-8*y^4 -
      2.537025399109203*10^-8*x*y^4 - 1.724836234416615*10^-9*x^2*y^4 +
      5.25751403746528*10^-9*y^5 + 8.62418117208308*10^-10*x*y^5), 0]

gammahat/: gammahat[{c, d, -b, h13P2}][x_] =
   If[x > 0 && x < 0.07215061722742411,
    4.25221475288933*10^-13 - 5.760074345265753*10^-12*x +
     2.304597695105914*10^-11*x^2 +
     E^(69.98977980221105*(1 - 0.855698765545152/(1 - 2*x))*x)*
      (-(4.25221475288933*10^-13) + 3.55212317679095*10^-11*x -
        (2.546658566775118*10^-11*x)/(1 - 2*x) -
        1.382182201456082*10^-9*x^2 -
        (7.122003240814243*10^-10*x^2)/(1 - 2*x)^2 +
        (1.995313809034501*10^-9*x^2)/(1 - 2*x) +
        1.427946746085649*10^-8*x^3 -
        (1.616978209499364*10^-9*x^3)/(1 - 2*x)^3 +
        (1.322313525417583*10^-8*x^3)/(1 - 2*x)^2 -
        (2.546362687459446*10^-8*x^3)/(1 - 2*x) -
        1.473959683201978*10^-8*x^4 -
        (4.57036016302446*10^-9*x^4)/(1 - 2*x)^4 +
        (1.486508927474308*10^-8*x^4)/(1 - 2*x)^3 -
        (2.681103001673053*10^-8*x^4)/(1 - 2*x)^2 +
        (3.093795933229985*10^-8*x^4)/(1 - 2*x) +
        3.986851919636061*10^-9*x^5 -
        (1.811890491613085*10^-9*x^5)/(1 - 2*x)^5 +
        (5.45350829872881*10^-9*x^5)/(1 - 2*x)^4 -
        (6.770401775674696*10^-9*x^5)/(1 - 2*x)^3 +
        (7.967027657177268*10^-9*x^5)/(1 - 2*x)^2 -
        (8.80011649324424*10^-9*x^5)/(1 - 2*x) -
        (2.972140778378114*10^-10*x^6)/(1 - 2*x)^5 +
        (1.389339752750309*10^-9*x^6)/(1 - 2*x)^4 -
        (2.435447745209466*10^-9*x^6)/(1 - 2*x)^3 +
        (1.897433881542712*10^-9*x^6)/(1 - 2*x)^2 -
        (5.543521733182261*10^-10*x^6)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, d, -b, h13P2}][] = 2.580459907366552*10^-17

gammahat/: gammahat[{c, d, -b, h13P1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.07215061722742411 &&
     y < 2*(1 - 0.855698765545152/(1 - 2*x))*x,
    0.821488769971851*E^(34.99488990110552*y)*
```

```
    (1.46406065580464*10^-10*x - 2.029172739007644*10^-9*x^2 +
      1.992041561699792*10^-10*y + 1.057266794170133*10^-8*x*y +
      6.937723678205484*10^-8*x^2*y - 1.320311198717069*10^-8*y^2 -
      1.532170895688915*10^-7*x*y^2 - 1.777101824177043*10^-7*x^2*y^2 +
      8.66552357913198*10^-8*y^3 + 8.88550912088521*10^-8*x*y^3), 0]

gammahat/: gammahat[{c, d, -b, h13P1}][x_] =
   If[x > 0 && x < 0.07215061722742411,
    9.24586086947788*10^-13 + 9.8212211219551*10^-12*x +
     1.009849326244766*10^-10*x^2 +
     E^(69.98977980221105*(1 - 0.855698765545152/(1 - 2*x))*x)*
      (-(9.24586086947788*10^-13) + 5.489035551170853*10^-11*x -
        (5.537361624190647*10^-11*x)/(1 - 2*x) -
        1.110344496621086*10^-9*x^2 -
        (1.418520685151054*10^-9*x^2)/(1 - 2*x)^2 +
        (2.521441438113403*10^-9*x^2)/(1 - 2*x) +
        4.905512352961863*10^-9*x^3 -
        (1.019634475085616*10^-8*x^3)/(1 - 2*x)^3 +
        (2.468940671339799*10^-8*x^3)/(1 - 2*x)^2 -
        (1.912531897082869*10^-8*x^3)/(1 - 2*x) -
        (1.045519217114609*10^-8*x^4)/(1 - 2*x)^3 +
        (2.443661856748212*10^-8*x^4)/(1 - 2*x)^2 -
        (1.42787506254692*10^-8*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, d, -b, h13P1}][] = 8.65324915081731*10^-16

gammahat/: gammahat[{c, d, -b, h13P0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.07949553454464048 &&
     y < 2*(1 - 0.841008930910719/(1 - 2*x))*x,
    1.672185923246665*10^-10*E^(34.99488990110552*y)*
     (-4*x^2 - y + 2*x*(0.158991069089281 + y)), 0]

gammahat/: gammahat[{c, d, -b, h13P0}][x_] =
   If[x > 0 && x < 0.07949553454464048,
    -(1.365448422811323*10^-13) - 1.246347172672338*10^-12*x +
     1.911348688876817*10^-11*x^2 +
     E^(69.98977980221105*(1 - 0.841008930910719/(1 - 2*x))*x)*
      (1.365448422811323*10^-13 - 8.31039627171175*10^-12*x +
        (8.03730658714948*10^-12*x)/(1 - 2*x) -
        (1.607461317429896*10^-11*x^2)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, d, -b, h13P0}][] = 1.651337522145818*10^-16

gammahat/: gammahat[{c, d, -b, h21S0}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.05727000551981654 &&
     y < 2*(1 - 0.885459988960367/(1 - 2*x))*x,
    1.549388990644905*10^-8*E^(33.05874942611109*y)*
     (0.3960795430418675 + 0.0443355162071467 2*
```

```
        (0.1368999999999994 - 39.31289999999999*y))^2*
    (-4*x^2 - y + 2*x*(0.1145400110396331 + y)), 0]

gammahat/: gammahat[{c, d, -b, h21S0}][x_] =
   If[x > 0 && x < 0.05727000551981654,
    -(3.731588660439022*10^-12) - 1.504993723723643*10^-11*x +
     3.931048085952164*10^-10*x^2 +
     E^(66.11749885222218*(1 - 0.885459988960367/(1 - 2*x))*x)*
      (3.731588660439022*10^-12 - 2.316733717363059*10^-10*x +
        (2.18463618439978*10^-10*x)/(1 - 2*x) + 2.904757359721873*10^-9*x^2 +
        (2.46572556678749*10^-9*x^2)/(1 - 2*x)^2 -
        (5.704808211864446*10^-9*x^2)/(1 - 2*x) -
        1.04302638815001*10^-8*x^3 +
        (7.907597565233198*10^-9*x^3)/(1 - 2*x)^3 -
        (3.070004222493607*10^-8*x^3)/(1 - 2*x)^2 +
        (3.382115915774233*10^-8*x^3)/(1 - 2*x) -
        (1.58151951304664*10^-8*x^4)/(1 - 2*x)^3 +
        (3.57219870522558*10^-8*x^4)/(1 - 2*x)^2 -
        (2.017142925576883*10^-8*x^4)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, d, -b, h21S0}][] = 4.476891840068346*10^-16

gammahat/: gammahat[{c, d, -b, h23S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.05275621996850904 &&
     y < 2*(1 - 0.894487560062982/(1 - 2*x))*x,
    0.907365798870541*E^(33.05874942611109*y)*
     (1.841401464022909*10^-10*x - 3.49039689561168*10^-9*x^2 +
       1.613395688055339*10^-9*y - 3.938613546792648*10^-8*x*y +
       1.452808221641166*10^-7*x^2*y - 3.378379430841943*10^-8*y^2 +
       3.70893711598461*10^-7*x*y^2 - 9.40531816129536*10^-7*x^2*y^2 +
       1.821037898962439*10^-7*y^3 - 7.051089502948175*10^-7*x*y^3 +
       0.000001814121524298627*x^2*y^3 - 2.308438661985126*10^-7*y^4 -
       9.03714987054903*10^-7*x*y^4 - 6.341953795035716*10^-8*x^2*y^4 +
       2.087187543046866*10^-7*y^5 + 3.170976897517858*10^-8*x*y^5), 0]

gammahat/: gammahat[{c, d, -b, h23S1}][x_] =
   If[x > 0 && x < 0.05275621996850904,
    4.011225200156241*10^-12 - 5.909698342024878*10^-11*x +
     2.719665822860897*10^-10*x^2 +
     E^(66.11749885222218*(1 - 0.894487560062982/(1 - 2*x))*x)*
      (-(4.011225200156241*10^-12) + 3.243091609875838*10^-10*x -
        (2.372289936111958*10^-10*x)/(1 - 2*x) - 9.68484916183385*10^-9*x^2 -
        (4.672392889872806*10^-9*x^2)/(1 - 2*x)^2 +
        (1.364324787022986*10^-8*x^2)/(1 - 2*x) +
        1.052546962552534*10^-7*x^3 -
        (3.360684775001112*10^-8*x^3)/(1 - 2*x)^3 +
        (1.50037897171249*10^-7*x^3)/(1 - 2*x)^2 -
        (2.198822742272793*10^-7*x^3)/(1 - 2*x) -
```

```
        3.673369051324314*10^-7*x^4 -
        (7.377282523870096*10^-8*x^4)/(1 - 2*x)^4 +
        (4.234315223448138*10^-7*x^4)/(1 - 2*x)^3 -
        (9.64056207490966*10^-7*x^4)/(1 - 2*x)^2 +
        (9.80214977401464*10^-7*x^4)/(1 - 2*x) + 1.84367056464602*10^-7*x^5 -
        (1.049733271675042*10^-7*x^5)/(1 - 2*x)^5 +
        (3.313659222136146*10^-7*x^5)/(1 - 2*x)^4 -
        (4.561154336501809*10^-7*x^5)/(1 - 2*x)^3 +
        (5.11595286152234*10^-7*x^5)/(1 - 2*x)^2 -
        (4.65816665752485*10^-7*x^5)/(1 - 2*x) -
        (1.594815935025276*10^-8*x^6)/(1 - 2*x)^5 +
        (7.131752329402919*10^-8*x^6)/(1 - 2*x)^4 -
        (1.195950505264841*10^-7*x^6)/(1 - 2*x)^3 +
        (8.91348714550137*10^-8*x^6)/(1 - 2*x)^2 -
        (2.491227252191679*10^-8*x^6)/(1 - 2*x)), 0]

gammahat/: gammahat[{c, d, -b, h23S1}][] = 2.642034420436096*10^-16

gammahatL/: gammahatL[{-b, -c, c, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3777754884528997 &&
     y < 2*(1 - 0.2444490230942007/(1 - 2*x))*x,
    7.832370089756881*10^-10*E^(4.779215211735137*y)*
     (-4*x^2 - y + 2*x*(0.7555509769057993 + y))*
     (0.0416142900683627*(-9.61 +
          0.006359235772481807*(9.61 + 39.31289999999999*(1 - y))^2) -
       0.4889875006166424*(-1 + 4.09083246618106*(1 - y)) +
       (5.876310404539604*(-0.4944178628389155 + 2.02258064516129*(1 - y))^2)/
        (5.610513777983813 - 24.46144999999999*y + 9.828225*y^2)), 0]

gammahatL/: gammahatL[{-b, -c, c, h13S1}][] = 2.155423357162919*10^-13

gammahatL/: gammahatL[{-b, -u, c, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4486161031111925 &&
     y < 2*(1 - 0.102767793777615/(1 - 2*x))*x,
    1.8179515952457*10^-11*E^(10.79844153223308*y)*
     (-4*x^2 - y + 2*x*(0.897232206222385 + y))*
     (0.1073044543538621*(-4.040099999999999 +
          0.006359235772481807*(4.040099999999999 +
              39.31289999999999*(1 - y))^2) -
       0.2803674038807863*(-1 + 9.73067498329249*(1 - y)) +
       (1.782051560529122*(-0.3205741626794258 + 3.119402985074627*(1 - y))^
           2)/(7.911973040910235 - 21.6765*y + 9.828225*y^2)), 0]

gammahatL/: gammahatL[{-b, -u, c, h13S1}][] = 6.3504866010001*10^-15

gammahatL/: gammahatL[{c, s, -b, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.1222295989357183 &&
     y < 2*(1 - 0.7555408021285634/(1 - 2*x))*x,
```

```
    1.629697465527432*10^-9*E^(28.58526263869093*y)*
     (-4*x^2 - y + 2*x*(0.2444591978714364 + y))*
     (0.02064312674336894*(-29.7025 +
          0.006359235772481807*(29.7025 + 39.31289999999999*(1 - y))^2) -
       0.6708575633405633*(-1 + 1.323555256291558*(1 - y)) +
       (7.213852325067998*(-0.869218500797448 + 1.15045871559633*(1 - y))^2)/
        (0.5873376688059153 - 34.5077*y + 9.828225*y^2)), 0]

gammahatL/: gammahatL[{c, s, -b, h13S1}][] = 1.888146340403938*10^-14

gammahatL/: gammahatL[{c, d, -b, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.1386822137262832 &&
     y < 2*(1 - 0.7226355725474336/(1 - 2*x))*x,
    1.263998148403405*10^-9*E^(33.05874942611109*y)*
     (-4*x^2 - y + 2*x*(0.2773644274525664 + y))*
     (0.02479421637851598*(-28.4089 +
          0.006359235772481807*(28.4089 + 39.31289999999999*(1 - y))^2) -
       0.5955417868656626*(-1 + 1.383823379293109*(1 - y)) +
       (4.948741452276764*(-0.850079744816587 + 1.176360225140713*(1 - y))^2)/
        (0.756095429235689 - 33.86089999999999*y + 9.828225*y^2)), 0]

gammahatL/: gammahatL[{c, d, -b, h13S1}][] = 1.956238112029673*10^-14

gammahatLI/: gammahatLI[{-b, -c, c, h13S1}][x_, y_] =
   7.832370089756881*10^-10*E^(4.779215211735137*y)*
    ((-4*x^3)/3 - x*y + x^2*(0.7555509769057993 + y))*
    (0.0416142900683627*(-9.61 +
         0.006359235772481807*(9.61 + 39.31289999999999*(1 - y))^2) -
      0.4889875006166424*(-1 + 4.09083246618106*(1 - y)) +
      (5.876310404539604*(-0.4944178628389155 + 2.022580645161290*(1 - y))^2)/
       (5.610513777983813 - 24.46144999999999*y + 9.828225*y^2))

gammahatLI/: gammahatLI[{-b, -u, c, h13S1}][x_, y_] =
   1.8179515952457*10^-11*E^(10.79844153223308*y)*
    ((-4*x^3)/3 - x*y + x^2*(0.897232206222385 + y))*
    (0.1073044543538621*(-4.040099999999999 +
         0.006359235772481807*(4.040099999999999 + 39.31289999999999*(1 - y))^
           2) - 0.2803674038807863*(-1 + 9.73067498329249*(1 - y)) +
      (1.782051560529122*(-0.3205741626794258 + 3.119402985074627*(1 - y))^2)/
       (7.911973040910235 - 21.6765*y + 9.828225*y^2))

gammahatLI/: gammahatLI[{c, s, -b, h13S1}][x_, y_] =
   1.629697465527432*10^-9*E^(28.58526263869093*y)*
    ((-4*x^3)/3 - x*y + x^2*(0.2444591978714364 + y))*
    (0.02064312674336894*(-29.7025 +
         0.006359235772481807*(29.7025 + 39.31289999999999*(1 - y))^2) -
      0.6708575633405633*(-1 + 1.323555256291558*(1 - y)) +
      (7.213852325067998*(-0.869218500797448 + 1.15045871559633*(1 - y))^2)/
```

```
      (0.5873376688059153 - 34.5077*y + 9.828225*y^2))

gammahatLI/: gammahatLI[{c, d, -b, h13S1}][x_, y_] =
   1.263998148403405*10^-9*E^(33.05874942611109*y)*
    ((-4*x^3)/3 - x*y + x^2*(0.2773644274525664 + y))*
    (0.02479421637851598*(-28.4089 +
         0.006359235772481807*(28.4089 + 39.31289999999999*(1 - y))^2) -
      0.5955417868656626*(-1 + 1.383823379293109*(1 - y)) +
      (4.948741452276764*(-0.850079744816587 + 1.176360225140713*(1 - y))^2)/
       (0.756095429235689 - 33.86089999999999*y + 9.828225*y^2))

gammahatLX/: gammahatLX[proc_][xlb_, y_] :=
   ((gammahatLI[proc][(#1 + #3)/4, #2] -
         gammahatLI[proc][Max[xlb, (#1 - #3)/4], #2] & )[2*#1 + #2, #2,
      Sqrt[(2*#1 + #2)^2 - 4*#2]] & )[xupper[proc], y]

gammahatT/: gammahatT[{-b, -c, c, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.3777754884528997 &&
     y < 2*(1 - 0.2444490230942007/(1 - 2*x))*x,
    1.32865060360022*10^-9*E^(4.779215211735137*y)*
     (0.007497514835448084*(94.0209664726337 +
          3.743839758191065*(-9.61 +
             0.006359235772481807*(9.61 + 39.31289999999999*(1 - y))^2))*y +
       0.881974149419067*y*(0.7555509769057993 - 4*x + y) +
       0.5894981019489739*(-4*x^2 - y + 2*x*(0.7555509769057993 + y))*
        (2.44591484059921 - 0.935959939547766*y -
          (5.876310404539604*(-0.4944178628389155 + 2.02258064516129*(1 - y))^
              2)/(5.610513777983813 - 24.46144999999999*y + 9.828225*y^2))),
    0]

gammahatT/: gammahatT[{-b, -c, c, h13S1}][] = 2.519575208987715*10^-13

gammahatT/: gammahatT[{-b, -u, c, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.4486161031111925 &&
     y < 2*(1 - 0.102767793777615/(1 - 2*x))*x,
    1.32865060360022*10^-9*E^(10.79844153223308*y)*
     (0.0001740229189796141*(28.51282496846595 +
          6.812801360061922*(-4.040099999999999 +
             0.006359235772481807*
              (4.040099999999999 + 39.31289999999999*(1 - y))^2))*y +
       0.01520747030607316*y*(0.897232206222385 - 4*x + y) +
       0.01368269122310734*(-4*x^2 - y + 2*x*(0.897232206222385 + y))*
        (1.764363813300782 - 1.70320034001548*y -
          (1.782051560529122*(-0.3205741626794258 +
                3.119402985074627*(1 - y))^2)/
           (7.911973040910235 - 21.6765*y + 9.828225*y^2))), 0]

gammahatT/: gammahatT[{-b, -u, c, h13S1}][] = 2.682561104662831*10^-15
```

```
gammahatT/: gammahatT[{c, s, -b, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.1222295989357183 &&
     y < 2*(1 - 0.7555408021285634/(1 - 2*x))*x,
    1.32865060360022*10^-9*E^(28.58526263869093*y)*
     (0.01560023439273327*(115.421637201088 +
          23.44752714050477*(-29.7025 +
             0.006359235772481807*(29.7025 + 39.31289999999999*(1 - y))^2))*y\
         + 5.088512002439632*y*(0.2444591978714364 - 4*x + y) +
       1.226580909316167*(-4*x^2 - y + 2*x*(0.2444591978714364 + y))*
        (0.97148082822227 - 5.861881785126192*y -
          (7.213852325067998*(-0.869218500797448 + 1.15045871559633*(1 - y))^
              2)/(0.5873376688059153 - 34.5077*y + 9.828225*y^2))), 0]

gammahatT/: gammahatT[{c, s, -b, h13S1}][] = 1.489526587820751*10^-14

gammahatT/: gammahatT[{c, d, -b, h13S1}][x_, y_] =
   If[x > 0 && y > 0 && x < 0.1386822137262832 &&
     y < 2*(1 - 0.7226355725474336/(1 - 2*x))*x,
    1.32865060360022*10^-9*E^(33.05874942611109*y)*
     (0.01209958768678103*(79.17986323642822 +
          22.30777160135291*(-28.4089 +
             0.006359235772481807*(28.4089 + 39.31289999999999*(1 - y))^2))*y\
         + 3.188405725996395*y*(0.2773644274525664 - 4*x + y) +
       0.951339761543308*(-4*x^2 - y + 2*x*(0.2773644274525664 + y))*
        (0.6967874788924265 - 5.576942900338229*y -
          (4.948741452276764*(-0.850079744816587 + 1.176360225140713*(1 - y))^
              2)/(0.756095429235689 - 33.86089999999999*y + 9.828225*y^2))),
    0]

gammahatT/: gammahatT[{c, d, -b, h13S1}][] = 1.573102473197147*10^-14

gammahatTI/: gammahatTI[{-b, -c, c, h13S1}][x_, y_] =
   1.32865060360022*10^-9*E^(4.779215211735137*y)*
    (0.007497514835448084*x*(94.0209664726337 +
         3.743839758191065*(-9.61 +
            0.006359235772481807*(9.61 + 39.31289999999999*(1 - y))^2))*y +
      0.881974149419067*y*(0.7555509769057993*x - 2*x^2 + x*y) +
      0.5894981019489739*((-4*x^3)/3 - x*y + x^2*(0.7555509769057993 + y))*
       (2.44591484059921 - 0.935959939547766*y -
         (5.876310404539604*(-0.4944178628389155 + 2.02258064516129*(1 - y))^
             2)/(5.610513777983813 - 24.46144999999999*y + 9.828225*y^2)))

gammahatTI/: gammahatTI[{-b, -u, c, h13S1}][x_, y_] =
   1.32865060360022*10^-9*E^(10.79844153223308*y)*
    (0.0001740229189796141*x*(28.51282496846595 +
         6.812801360061922*(-4.040099999999999 +
            0.006359235772481807*
```

```
            (4.040099999999999 + 39.31289999999999*(1 - y))^2))*y +
      0.01520747030607316*y*(0.897232206222385*x - 2*x^2 + x*y) +
      0.01368269122310734*((-4*x^3)/3 - x*y + x^2*(0.897232206222385 + y))*
       (1.764363813300782 - 1.70320034001548*y -
         (1.782051560529122*(-0.3205741626794258 + 3.119402985074627*(1 - y))^
            2)/(7.911973040910235 - 21.6765*y + 9.828225*y^2)))

gammahatTI/: gammahatTI[{c, s, -b, h13S1}][x_, y_] =
   1.32865060360022*10^-9*E^(28.58526263869093*y)*
    (0.01560023439273327*x*(115.421637201088 +
         23.44752714050477*(-29.7025 +
            0.006359235772481807*(29.7025 + 39.31289999999999*(1 - y))^2))*y \
       + 5.088512002439632*y*(0.2444591978714364*x - 2*x^2 + x*y) +
      1.226580909316167*((-4*x^3)/3 - x*y + x^2*(0.2444591978714364 + y))*
       (0.97148082822227 - 5.861881785126192*y -
         (7.213852325067998*(-0.869218500797448 + 1.15045871559633*(1 - y))^
            2)/(0.5873376688059153 - 34.5077*y + 9.828225*y^2)))

gammahatTI/: gammahatTI[{c, d, -b, h13S1}][x_, y_] =
   1.32865060360022*10^-9*E^(33.05874942611109*y)*
    (0.01209958768678103*x*(79.17986323642822 +
         22.30777160135291*(-28.4089 +
            0.006359235772481807*(28.4089 + 39.31289999999999*(1 - y))^2))*y \
       + 3.188405725996395*y*(0.2773644274525664*x - 2*x^2 + x*y) +
      0.951339761543308*((-4*x^3)/3 - x*y + x^2*(0.2773644274525664 + y))*
       (0.6967874788924265 - 5.576942900338229*y -
         (4.948741452276764*(-0.850079744816587 + 1.176360225140713*(1 - y))^
            2)/(0.756095429235689 - 33.86089999999999*y + 9.828225*y^2)))

gammahatTX/: gammahatTX[proc_][xlb_, y_] :=
   ((gammahatTI[proc][(#1 + #3)/4, #2] -
          gammahatTI[proc][Max[xlb, (#1 - #3)/4], #2] & )[2*#1 + #2, #2,
       Sqrt[(2*#1 + #2)^2 - 4*#2]] & )[xupper[proc], y]

xupper/: xupper[{-b, -c, c, h11S0}] = 0.3870548852921051

xupper/: xupper[{-b, -c, c, h13S1}] = 0.3777754884528997

xupper/: xupper[{-b, -c, c, h11P1}] = 0.3433071587188938

xupper/: xupper[{-b, -c, c, h13P2}] = 0.3388111790277492

xupper/: xupper[{-b, -c, c, h13P1}] = 0.3433071587188938

xupper/: xupper[{-b, -c, c, h13P0}] = 0.3512396694214875

xupper/: xupper[{-b, -c, c, h21S0}] = 0.3360830668813544
```

```
xupper/: xupper[{-b, -c, c, h23S1}] = 0.3268240195966209

xupper/: xupper[{-b, -u, c, h11S0}] = 0.4559991758430439

xupper/: xupper[{-b, -u, c, h13S1}] = 0.4486161031111925

xupper/: xupper[{-b, -u, c, h11P1}] = 0.4242793078099046

xupper/: xupper[{-b, -u, c, h13P2}] = 0.4205095528439774

xupper/: xupper[{-b, -u, c, h13P1}] = 0.4211442045740711

xupper/: xupper[{-b, -u, c, h13P0}] = 0.4267416039010096

xupper/: xupper[{-b, -u, c, h21S0}] = 0.4153407660081042

xupper/: xupper[{-b, -u, c, h23S1}] = 0.4113573407202216

xupper/: xupper[{c, s, -b, h11S0}] = 0.1305016928285626

xupper/: xupper[{c, s, -b, h13S1}] = 0.1222295989357183

xupper/: xupper[{c, s, -b, h11P1}] = 0.06026647741581009

xupper/: xupper[{c, s, -b, h13P2}] = 0.06026647741581009

xupper/: xupper[{c, s, -b, h13P1}] = 0.06026647741581009

xupper/: xupper[{c, s, -b, h13P0}] = 0.06771314250538618

xupper/: xupper[{c, s, -b, h21S0}] = 0.04518237016348309

xupper/: xupper[{c, s, -b, h23S1}] = 0.04060753594875977

xupper/: xupper[{c, d, -b, h11S0}] = 0.1454293628808864

xupper/: xupper[{c, d, -b, h13S1}] = 0.1386822137262832

xupper/: xupper[{c, d, -b, h11P1}] = 0.07215061722742411

xupper/: xupper[{c, d, -b, h13P2}] = 0.07215061722742411

xupper/: xupper[{c, d, -b, h13P1}] = 0.07215061722742411

xupper/: xupper[{c, d, -b, h13P0}] = 0.07949553454464048

xupper/: xupper[{c, d, -b, h21S0}] = 0.05727000551981654
```

```
xupper/: xupper[{c, d, -b, h23S1}] = 0.05275621996850904

yupper/: yupper[{-b, -c, c, h11S0}][x_] =
   2*(1 - 0.2258902294157897/(1 - 2*x))*x

yupper/: yupper[{-b, -c, c, h11S0}][] = 0.2753320156996813

yupper/: yupper[{-b, -c, c, h13S1}][x_] =
   2*(1 - 0.2444490230942007/(1 - 2*x))*x

yupper/: yupper[{-b, -c, c, h13S1}][] = 0.2556132974163696

yupper/: yupper[{-b, -c, c, h11P1}][x_] =
   2*(1 - 0.3133856825622124/(1 - 2*x))*x

yupper/: yupper[{-b, -c, c, h11P1}][] = 0.1937684576818296

yupper/: yupper[{-b, -c, c, h13P2}][x_] =
   2*(1 - 0.3223776419445017/(1 - 2*x))*x

yupper/: yupper[{-b, -c, c, h13P2}][] = 0.1868114537467345

yupper/: yupper[{-b, -c, c, h13P1}][x_] =
   2*(1 - 0.3133856825622124/(1 - 2*x))*x

yupper/: yupper[{-b, -c, c, h13P1}][] = 0.1937684576818296

yupper/: yupper[{-b, -c, c, h13P0}][x_] =
   2*(1 - 0.2975206611570249/(1 - 2*x))*x

yupper/: yupper[{-b, -c, c, h13P0}][] = 0.2066115702479338

yupper/: yupper[{-b, -c, c, h21S0}][x_] =
   2*(1 - 0.3278338662372912/(1 - 2*x))*x

yupper/: yupper[{-b, -c, c, h21S0}][] = 0.1826983000490932

yupper/: yupper[{-b, -c, c, h23S1}][x_] =
   2*(1 - 0.3463519608067582/(1 - 2*x))*x

yupper/: yupper[{-b, -c, c, h23S1}][] = 0.1693184679837915

yupper/: yupper[{-b, -u, c, h11S0}][x_] =
   2*(1 - 0.0880016483139123/(1 - 2*x))*x

yupper/: yupper[{-b, -u, c, h11S0}][] = 0.4947002129072137

yupper/: yupper[{-b, -u, c, h13S1}][x_] =
```

```
   2*(1 - 0.102767793777615/(1 - 2*x))*x

yupper/: yupper[{-b, -u, c, h13S1}][] = 0.4616194684187633

yupper/: yupper[{-b, -u, c, h11P1}][x_] =
   2*(1 - 0.1514413843801907/(1 - 2*x))*x

yupper/: yupper[{-b, -u, c, h11P1}][] = 0.3731319744918335

yupper/: yupper[{-b, -u, c, h13P2}][x_] =
   2*(1 - 0.1589808943120452/(1 - 2*x))*x

yupper/: yupper[{-b, -u, c, h13P2}][] = 0.3615327284428266

yupper/: yupper[{-b, -u, c, h13P1}][x_] =
   2*(1 - 0.1577115908518578/(1 - 2*x))*x

yupper/: yupper[{-b, -u, c, h13P1}][] = 0.363453217646116

yupper/: yupper[{-b, -u, c, h13P0}][x_] =
   2*(1 - 0.1465167921979808/(1 - 2*x))*x

yupper/: yupper[{-b, -u, c, h13P0}][] = 0.380966552963531

yupper/: yupper[{-b, -u, c, h21S0}][x_] =
   2*(1 - 0.1693184679837916/(1 - 2*x))*x

yupper/: yupper[{-b, -u, c, h21S0}][] = 0.3463519608067581

yupper/: yupper[{-b, -u, c, h23S1}][x_] =
   2*(1 - 0.1772853185595568/(1 - 2*x))*x

yupper/: yupper[{-b, -u, c, h23S1}][] = 0.3351800554016621

yupper/: yupper[{c, s, -b, h11S0}][x_] =
   2*(1 - 0.7389966143428748/(1 - 2*x))*x

yupper/: yupper[{c, s, -b, h11S0}][] = 0.01969836872883964

yupper/: yupper[{c, s, -b, h13S1}][x_] =
   2*(1 - 0.7555408021285634/(1 - 2*x))*x

yupper/: yupper[{c, s, -b, h13S1}][] = 0.01710380053366705

yupper/: yupper[{c, s, -b, h11P1}][x_] = 2*(1 - 0.87946704516838/(1 - 2*x))*x

yupper/: yupper[{c, s, -b, h11P1}][] = 0.00386895904397793
```

```
yupper/: yupper[{c, s, -b, h13P2}][x_] = 2*(1 - 0.87946704516838/(1 - 2*x))*x

yupper/: yupper[{c, s, -b, h13P2}][] = 0.00386895904397793

yupper/: yupper[{c, s, -b, h13P1}][x_] = 2*(1 - 0.87946704516838/(1 - 2*x))*x

yupper/: yupper[{c, s, -b, h13P1}][] = 0.00386895904397793

yupper/: yupper[{c, s, -b, h13P0}][x_] = 2*(1 - 0.864573714989228/(1 - 2*x))*x

yupper/: yupper[{c, s, -b, h13P0}][] = 0.004924592182209895

yupper/: yupper[{c, s, -b, h21S0}][x_] = 2*(1 - 0.909635259673034/(1 - 2*x))*x

yupper/: yupper[{c, s, -b, h21S0}][] = 0.002139246913862868

yupper/: yupper[{c, s, -b, h23S1}][x_] = 2*(1 - 0.91878492810248/(1 - 2*x))*x

yupper/: yupper[{c, s, -b, h23S1}][] = 0.001719537352879077

yupper/: yupper[{c, d, -b, h11S0}][x_] =
   2*(1 - 0.7091412742382273/(1 - 2*x))*x

yupper/: yupper[{c, d, -b, h11S0}][] = 0.02493074792243765

yupper/: yupper[{c, d, -b, h13S1}][x_] =
   2*(1 - 0.7226355725474336/(1 - 2*x))*x

yupper/: yupper[{c, d, -b, h13S1}][] = 0.02247608291425968

yupper/: yupper[{c, d, -b, h11P1}][x_] = 2*(1 - 0.855698765545152/(1 - 2*x))*x

yupper/: yupper[{c, d, -b, h11P1}][] = 0.005619020728564927

yupper/: yupper[{c, d, -b, h13P2}][x_] = 2*(1 - 0.855698765545152/(1 - 2*x))*x

yupper/: yupper[{c, d, -b, h13P2}][] = 0.005619020728564927

yupper/: yupper[{c, d, -b, h13P1}][x_] = 2*(1 - 0.855698765545152/(1 - 2*x))*x

yupper/: yupper[{c, d, -b, h13P1}][] = 0.005619020728564927

yupper/: yupper[{c, d, -b, h13P0}][x_] = 2*(1 - 0.841008930910719/(1 - 2*x))*x

yupper/: yupper[{c, d, -b, h13P0}][] = 0.006878149411516307

yupper/: yupper[{c, d, -b, h21S0}][x_] = 2*(1 - 0.885459988960367/(1 - 2*x))*x
```

```
yupper/: yupper[{c, d, -b, h21S0}][] = 0.003482317509011026

yupper/: yupper[{c, d, -b, h23S1}][x_] = 2*(1 - 0.894487560062982/(1 - 2*x))*x

yupper/: yupper[{c, d, -b, h23S1}][] = 0.002940510621195586

mB = 6.27*GeV
```

# References


[1] P. Franzini, in *Proceedings of the XXIV International Conference on High Energy Physics*, Munich, 1988, edited by R. Kotthaus and J. H. Kuhn (Springer-Verlag, Berlin, 1989), p. 891.

[2] Particle Data Group, Phys. Lett. B **204**, 1 (1988).

[3] D. J. Gross and F. Wilczek, Phys. Rev. Lett. **30**, 1343 (1973).

[4] H. D. Politzer, Phys. Rev. Lett. **30**, 1346 (1973).

[5] N. Isgur and C. H. L. Smith, Nuclear Physics **B317**, 526 (1989).

[6] D. Griffiths, *Introduction to Elementary Particles*, (John Wiley & Sons, New York, 1987), p. 184.

[7] S. Gasiorowicz and J. L. Rosner, Am. J. Phys. **49**, 954 (1981).

[8] T. Applequist and H. D. Politzer, Phys. Rev. Lett. **34**, 43 (1975).

[9] D. H. Perkins, *Introduction to High Energy Physics, 3rd Ed.*, (Addison-Wesley, Menlo Park, 1987), p. 297.

[10] S. Godfrey and N. Isgur, Phys. Rev. D **32**, 189 (1985).

[11] E. Eichten and K. Gottfried, Phys. Lett. B **66B**, 286 (1977).

[12] C. Hayne and N. Isgur, Phys. Rev. D **25**, 1944 (1982).

[13] S. Capstick, S. Godfrey, N. Isgur, and J. Paton, Phys. Lett. B **175**, 457 (1986).

[14] G. Altarelli, N. Cabbibo, G. Corbò, L. Maiani, and G. Martinelli, Nucl. Phys. **B208**, 365 (1982).

[15] J. G. Körner and G. A. Schuler, Z. Phys. C **38**, 511 (1988).

[16] M. Wirbel, B. Stech, and M. Bauer, Z. Phys. C **29**, 637 (1985).

[17] N. Isgur, D. Scora, B. Grinstein, and M. Wise, Phys. Rev. D **39**, 799 (1989).

[18] D. Du and Z. Wang, Phys. Rev. D **39**, 1342 (1989).

[19] B. Grinstein and M. B. Wise, Phys. Lett. B **197**, 249 (1987).

[20] T. Altomari and L. Wolfenstein, Phys. Rev. D **37**, 681 (1988).

[21] D. Scora and N. Isgur, Phys. Rev. D **40**, 1491 (1989).


**Final Note:** This manuscript was prepared entirely on a NeXT computer, using TeX, LaTeX, METAFONT, PostScript, *Mathematica*, and GNUPLOT, and was printed on a 400 dpi NeXT Laser Printer.